UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique et Mathématique Appliquée**

Bourse: Colombienne

Présentée par

Julián QUIROGA SEPULVEDA

Thèse dirigée par **James L. CROWLEY**
et codirigée par **Frédéric DEVERNAY**

préparée au sein **du Laboratoire d'Informatique de Grenoble á l'INRIA Rhône-Alpes**
et de **l'Ecole Doctorale de Mathématiques, Sciences et Technologies de l'Information**

Scene Flow Estimation from RGBD Images

Thèse soutenue publiquement le **7 novembre 2014**,
devant le jury composé de :

M. Peter STURM

Research Director at INRIA Grenoble, Président

Mme. Lourdes AGAPITO

Reader at University College London, Rapporteur

M. Pierre KORNPROBST

Researcher at INRIA Sophia Antipolis, Rapporteur

M. Thomas BROX

Professor at University of Freiburg, Examinateur

M. James L. CROWLEY

Professor at INPG-ENSIMAG, Directeur de thèse

M. Frédéric DEVERNAY

Researcher at INRIA Grenoble, Co-Directeur de thèse

To Myriam and German who supported me from heaven and earth.

Acknowledgments

I would like to thank my advisors Frederic Dévérnay and James Crowley for the opportunity of doing this thesis under their supervision. I also would like to thank Thomas Brox for the interesting discussions and for providing a nice research environment at University of Freiburg.

I would like to especially thank Sergi Pujades and Laurent Boiron for their unconditional support and encouragement while this rough road trip. I must also specially thank INRIA Grenoble for providing a wonderful research environment and for supporting me in every research activity while this period.

This project would have been impossible without the support of the Pontificia Universidad Javeriana and Colciencias. Also I am grateful for the support of Campus France, Region Rhones-Alpes and GdR ISIS.

My finally thanks are reserved for my family and friends in Colombia who were my main motivation to achieve this thesis.

Julián Quiroga Sepúlveda
Grenoble, 13th November 2014

Scene Flow Estimation from RGBD Images

Abstract:

This thesis addresses the problem of reliably recovering a 3D motion field, or scene flow, from a temporal pair of RGBD images. We propose a semi-rigid estimation framework for the robust computation of scene flow, taking advantage of color and depth information, and an alternating variational minimization framework for recovering rigid and non-rigid components of the 3D motion field. Previous attempts to estimate scene flow from RGBD images have extended optical flow approaches without fully exploiting depth data or have formulated the estimation in 3D space disregarding the semi-rigidity of real scenes. We demonstrate that scene flow can be robustly and accurately computed in the image domain by solving for 3D motions consistent with color and depth, encouraging an adjustable combination between local and piecewise rigidity. Additionally, we show that solving for the 3D motion field can be seen as a specific case of a more general estimation problem of a 6D field of rigid motions. Accordingly, we formulate scene flow estimation as the search of an optimal field of twist motions achieving state-of-the-art results.

Keywords: motion, depth, scene flow, rgbd, variational, semi-rigid

Estimation du Flot de Scène à partir des Images RGBD

Résumé:

Cette thèse aborde le problème du calcul de manière fiable d'un champ de mouvement 3D, appelé flot de scène, à partir d'une paire d'images RGBD prises à des instants différents. Nous proposons un schéma d'estimation semi-rigide pour le calcul robuste du flot de scène, en prenant compte de l'information de couleur et de profondeur, et un cadre de minimisation alternée variationnelle pour récupérer les composantes rigides et non rigides du champ de mouvement 3D. Les tentatives précédentes pour estimer le flot de scène à partir des images RGBD étaient des extensions des approches de flux optique, et n'exploitaient pas totalement les données de profondeur, ou bien elles formulaient l'estimation dans l'espace 3D sans tenir compte de la semi-rigidité des scènes réelles. Nous démontrons que le flot de scène peut être calculé de manière robuste et précise dans le domaine de l'image en reconstruisant un mouvement 3D cohérent avec la couleur et la profondeur, en encourageant une combinaison réglable entre rigidité locale et par morceaux. En outre, nous montrons que le calcul du champ de mouvement 3D peut être considéré comme un cas particulier d'un problème d'estimation plus général d'un champ de mouvements rigides à 6 dimensions. L'estimation du flot de scène est donc formulée comme la recherche d'un champ optimal de mouvements rigides. Nous montrons finalement que notre méthode permet d'obtenir des résultats comparables à l'état de l'art.

Mots-Clés: mouvement, profondeur, flot de scène, rgbd, variationnelle, semi-rigide

Publications related with this thesis

1. **J. Quiroga**, F. Devernay and J. L. Crowley, "Scene Flow by Tracking in Intensity and Depth Data," in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Providence (USA), Jun. 2012.

2. **J. Quiroga**, F. Devernay and J. L. Crowley, "Local/global scene flow estimation," in *International Conference on Image Processing (ICIP)*, Melbourne (Australia), Sep. 2013.

3. **J. Quiroga**, F. Devernay and J. L. Crowley, "*Local Scene Flow by Tracking in Intensity and Depth*," Journal of Visual Communication and Image Representation, Jan. 2014.

4. **J. Quiroga**, T. Brox, F. Devernay and J. L. Crowley, "Dense Semi-Rigid Scene Flow Estimation From RGBD Images," in *European Conference on Computer Vision (ECCV)*, Zurich (Switzerland), Sep. 2014.

Notation

$\mathbb{R}$	set of real numbers
$\|a\| = \sqrt{a^2}$	absolute value, $a \in \mathbb{R}$
$\mathbf{a} = (a_1, ..., a_N)^T$	column vector
$\|\mathbf{a}\| = \sqrt{\sum {a_i}^2}$	Euclidean norm, or magnitude of the vector
$\|\mathbf{f}\|_p = (\sum \|f\|^p)^{\frac{1}{p}}$	l_p norm, f with countable domain
$\|\|\mathbf{f}\|\|_p = \left(\int \|f\|^p\right)^{\frac{1}{p}}$	L^p norm, f with uncountable domain
$\mathbf{\Omega}$	image domain
$\mathbf{x} = (x, y)^T$	image point
$I_c(\mathbf{x})$	color image
$I(\mathbf{x})$	brightness image
$Z(\mathbf{x})$	depth image
$\{I_c(\mathbf{x}), Z(\mathbf{x})\}$	RGBD image
$\nabla f = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})$	gradient operator
$I_g(\mathbf{x})$	magnitude of the gradient image
$\nabla \cdot f = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}$	divergence operator
$\mathbf{X} = (X, Y, Z)^T$	3D point
$\tilde{\mathbf{X}} = (X, Y, Z, 1)^T$	homogeneous 3D point
$\pi(\cdot)$	projective function
$\pi^{-1}(\cdot, Z(\cdot))$	inverse projective function
$\mathbf{u} = (u, v)^T$	2D motion vector
$\mathbf{v} = (v_X, v_Y, v_Z)^T$	3D motion vector
$\xi = (\omega, \tau)^T$	twist motion

$\omega = (\omega_X, \omega_Y, \omega_Z)^T$	3D rotation vector
$\tau = (\tau_X, \tau_Y, \tau_Z)^T$	3D translation vector
$\mathbf{u}(\mathbf{x})$	image flow, or 2D motion field
$\mathbf{v}(\mathbf{x})$	scene flow, or 3D motion field
$\xi(\mathbf{x})$	twist motion field, or field of rigid motions
$\omega(\mathbf{x})$	rotational field, or field of rotations
$\tau(\mathbf{x})$	translational field, or field of translations
$\{\mathbf{R}, \mathbf{t}\}$	Rigid body motion
$\mathbf{R}$	3×3 rotation matrix
$\mathbf{t}$	3D translation vector

Contents

1 Introduction

1.1 Thesis theme

In the real world everything changes. Changes occur for many reasons but nothing remains the same. Some of these changes are sudden while others are subtle. Some are induced while others are provoked, but always these changes reveal information that was hidden from our initial perception. Visual changes brought by relative motions of the environment are particularly meaningful, allowing the perception of, and interaction with, the surrounding world. For instance, a chaotic motion alerts us to possible dangers. A peculiar way of motion tells us that someone known is approaching. A subtle gesture of the hand is enough to communicate our dislike. Being able to understand visual changes is a wonderful ability of human beings, which inspires the development of computer vision systems aiming to improve the way we perceive, and we are perceived, in our environment.

A computer vision system sees the world through a camera, which usually corresponds to a color-sensitive retina providing a planar representation of the scene. Visual changes in the world are perceived as displacements of the color patterns on the retina, and the computation of such those motions enables understanding of a changing environment. Image motion estimate provides powerful cues for visual systems, and has been exploited in a wide variety of applications including autonomous navigation, action recognition, human-machine interaction, three-dimensional (3D) reconstruction. For example, by estimating and learning motion patterns, visual systems are able to recognize actions, activities and gestures. By using a single moving camera and tracking a set of key points it is possible to recover the 3D structure of a static scene. Even, the shape of a time-varying 3D surface can be estimated, after a sufficiently long observation. Nevertheless, this planar representation of the scene is only a projection of the 3D world, and therefore the perception and estimation of some motions may become a challenge. For instance, the lack of texture hinders motion estimation, because there is not enough information to disambiguate between all of the possible motions that can explain the observations. Moreover, it is complicated to handle with partial occlusions,

which can severely confound motion estimation. For such of challenges, motion estimation using monocular color images is still a very active area of research. There is a physical limitation due to the use of a single view, which prevents the reliable estimation of the full 3D motion field of the scene. If a fully calibrated stereo or multi-view camera system is available, it is possible to estimate both the motion and the structure of the scene. Evidently, when the number of views increases, there is more information available to deal with occlusions and motion ambiguities, but at the same time, such system becomes less affordable and portable.

Recently with the arrival of depth cameras, based either on time-of-flight (ToF) or structured light sensing, it has been possible to have direct measurements of the 3D structure of the scene. Such cameras are also known as RGBD (RGB and Depth) cameras, since they simultaneously provide color and depth information. Access to structure information opens the door to the estimation of motions directly in the 3D scene, using a single view. For this reason, current depth sensors have decreased the requirements of the system needed for this kind of tasks. Moreover, the availability of RGBD representation of the scene demands the reformulation of the 3D motion estimation problem. Usually, when only color information is available, motion is solved to be consistent with the displacement of color patterns. However, for RGBD cameras, both color and 3D structure can be simultaneously exploited. For instance, action recognition is usually based on features obtained from 2D trajectories computed on the image. The simple addition of the depth information to every trajectory significantly improves the recognition performance. However, it is clear that this way depth data is not fully exploited. The combined use of color and depth can yield more accurate trajectories, incorporating at the same time, the 3D structure information. In general, the use of depth data in addition to color can yield better results for computer vision tasks, such as recognition, interaction and reconstruction. Both color and depth data complement each other and can be combined as whole to obtain the most benefit from RGBD images.

Scene flow is defined as the motion field in 3D space, and can be computed from a single view when using an RGBD sensor. Being able to confidently estimate the scene flow makes it possible to directly take advantage of real world motion. In this thesis, we address the problem of scene flow estimation from RGBD images; by rethinking the way color and depth are exploited. The two main questions for scene flow estimation from RGBD images are: how to fully exploit the data?, and which motion model should be used?. We go beyond previous works and propose a scene flow framework that exploits the local and piecewise rigidity of real world scenes. By modeling the motion as a field of twists, our method encourages piecewise smooth solutions of rigid body motions. We give a general formulation to solve for local and global rigid motions by jointly using intensity and depth data. In order to deal efficiently with a moving camera, we model motion as a rigid component plus a non-rigid residual and propose an alternating solver. The evaluation demonstrates that use of the proposed framework achieves the best results in the most commonly used scene flow benchmark. Moreover, through additional experiments we indicate the general applicability of our approach in a variety of different scenarios.

1.2 Thesis outline

The following six chapters present the basics, methods and results for scene flow estimation from RGBD images.

Chapter 2. The first part of this chapter introduces two important concepts about the scene: the RGBD representation, based on color and depth data, and the scene flow, which models the motion observed in the scene and is the main concern of this thesis. The second part presents three useful tools for motion representation and estimation. Fundamentals of rigid body motion are reviewed, particularly, the twist motion representation that we use to model the motion in the scene. We summarize the concept of total variation, describing its role and properties as regularizer and presenting some basic solvers. Finally, we present the basics of robust least-squares estimation.

Chapter 3. The chapter analyses the estimation of 3D motion when color and structure of the scene are available. We first define the scene flow estimation problem, pointing out the main challenges of this task and analyzing properties of real scenes that can be exploited to aid the estimation. We then present a survey of the most relevant work done in scene flow estimation, where the advantages and limitations of most known techniques are studied. Simultaneously, we explain how the ideas supported by this thesis go beyond previous methods. Having presented the state of the art for scene flow estimation, we finally give a brief exposition of the main components of our approach.

Chapter 4. In this chapter, we present alternative representations for 3D motion in the image domain. First, we introduce the parameterization of the scene flow as a field of rigid motions, using a twist representation. Here we define a warping function to constrain every twist motion on the image, and which is also provided with a linear version allowing an incremental solution of rigid motion. Alternatively, we present the direct representation of the scene flow as a 3D motion field, using a warping function to exploit RGBD data. Finally, we review other possibilities for motion representation and we conclude with a table containing the three cases considered in this thesis.

Chapter 5. The framework for scene flow estimation from RGBD images is presented. We begin by motivating the semi-rigid constraint of the 3D motion field on the image and then we formulate the scene flow energy based on the twist motion representation. Subsequently, each of the components of this energy is described. By considering the 3D motion representation, we formulate an alternative scene flow energy, which is also described in detail. We then present the minimization of both energies and a coarse-to-fine procedure. Finally, we introduce an approach to estimate the scene flow with a moving camera using the proposed framework.

Chapter 6. This chapter presents the experiments performed using the proposed framework. The first part contains the motion estimation results on the Middlebury stereo dataset. Using this benchmark we compare the accuracy of our approach with the most relevant scene flow methods. Also, we analyze the different components our framework and the alternatives that it supports. The second part of the experimentation considers images from RGBD sensors. We investigate the performance of our method in different and interesting scenarios presenting non-rigid and rigid motions.

Chapter 7. We conclude by reviewing the formulated questions and describing how this thesis contributes to their resolution. We summarize the limitations of this approach and propose some directions for future work. Finally, we conclude by discussing the impact of our work, some of the lessons learned and perspectives on scene flow estimation from RGBD images.

2 Background

2.1 RGBD representation of a scene

Let $\mathbf{\Omega}$ be the image domain defined as a regular $L^x \times L^y$ grid of pixels, indexed as $\mathbf{x} = (x, y)$, for $x = 1, 2, ..., L^x$, $y = 1, 2, ..., L^y$. The use of a RGBD camera, e.g., Microsoft Kinect for Xbox or Asus Xtion Pro Live, enables color (RGB) and depth (D) discrete representations of the observed scene in the image domain. For every time t the scene is registered as a RGBD image $\mathbf{S}_t(\mathbf{x}) = \{I_c(\mathbf{x}), Z_t(\mathbf{x})\}$ where images $I(\mathbf{x})$ and $Z(\mathbf{x})$ provide color and depth, respectively, for every pixel $\mathbf{x} \in \mathbf{\Omega}$. Figure 2.1 shows a pair of color and depth images captured by a Kinect for Xbox.

Fig. 2.1: *RGBD representation. (a) RGB image and (b) depth image. The depth image is encoded in gray levels, where the closer a point is at the brighter is drawn. Pixels having no valid depth measures are drawn in black. Note that field of views of RGB and depth cameras are different.*

This kind of camera provides both RGB and depth with VGA resolution (640×480 pixels). The RGB image uses 8-bit pixel representation with a Bayer color filter. The depth image uses 11-bit pixels to encode the 3D structure of the scene, using *Light Coding* by PrimeSense. This technology codes the scene with near-IR light and with an infrared camera that captures distorted light to measure the relative

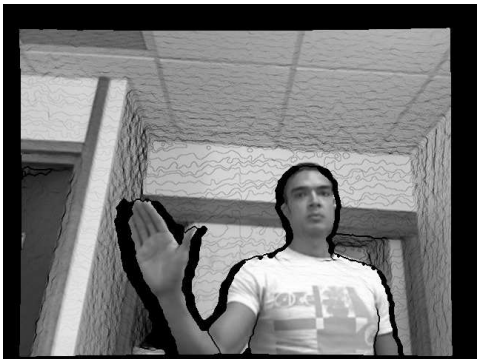


Fig. 2.2: *The RGBD image is displayed by showing the gray valued of every pixel having a valid depth measure. We also incorporate information about 3D structure as contour lines, which are defined as a function of the point-wise changes in depth data.*

distance of the surface. As is shown in Figure 2.1 b), some pixels do not have a valid depth measure for one of these reasons: *i*) a depth out of the sensor range ($80cm$ to $400cm$ for the Kinect for Xbox), *ii*) an occluded point in the triangulation between IR emitter and IR camera, *iii*) a point out of the field of view of the depth sensor but visible for the RGB camera, or *iv*) a poor or non reflective surface to IR light.

In this RGBD representation every pixel $\mathbf{x} \in \mathbf{\Omega}$, having valid depth measure, represents a 3D point $\mathbf{X}$ of the scene, or *scene point*. For illustration, in some cases we display the RGBD image as a single image, by blending gray values and depth data, as is described in Figure 2.2. Below we present the camera model used to define a bijective mapping between 3D space and image domain.

2.1.1 Camera model

The camera frame is defined with its origin at the optical center and its (X, Y, Z) axes respectively in the direction of the image axes and the optical axis, with positive Z pointing in the direction of sight, as is shown in Figure 2.3.

In the RGBD representation, every 3D point that is visible for the sensor, is parametrized into the image domain $\mathbf{\Omega}$. The projection $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ maps a 3D point $\mathbf{X} = (X, Y, Z)^T$ onto a pixel $\mathbf{x} = (x, y)^T$ on $\mathbf{\Omega}$ by:

$$\pi(\mathbf{X}) = \left(f_x \frac{X}{Z} + c_x, f_y \frac{Y}{Z} + c_y \right)^T, \tag{2.1}$$

where f_x and f_y are the focal lengths of the camera and (c_x, c_y) its principal point. The inverse projection $\pi^{-1} : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^3$ back-projects an image point to the 3D space for a given depth z, as follows:

$$\pi^{-1}(\mathbf{x}, z) = \left(z \frac{x - c_x}{f_x}, z \frac{y - c_y}{f_y}, z \right)^T. \tag{2.2}$$

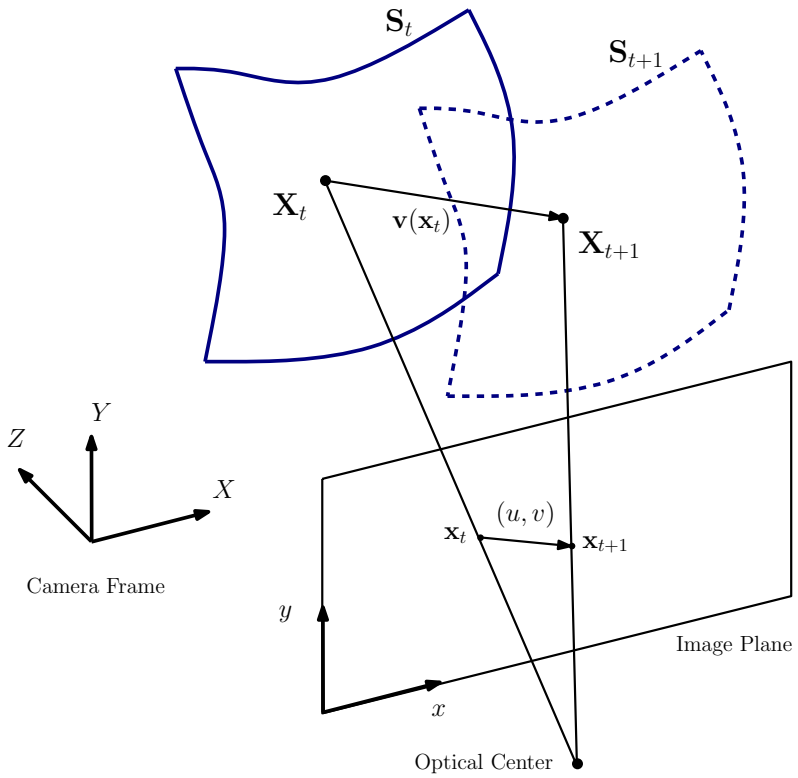


Fig. 2.3: *Motion of a rigid surface point in 3D space and in the image domain*

Given a RGBD image $\mathbf{S}(\mathbf{x}) = \{I_c(\mathbf{x}), Z(\mathbf{x})\}$, the notations $\mathbf{X}$ or $\pi^{-1}(\mathbf{x}, Z(\mathbf{x}))$ are equivalently used as the 3D representation of image point $\mathbf{x}$.

2.1.2 Scene flow definition

Let $\mathbf{X}_t = (X_t, Y_t, Z_t)^T$ be coordinates at time t of a scene point in the camera reference frame. The scene flow, $\mathbf{v}(\mathbf{x}) : \mathbf{\Omega} \rightarrow \mathbb{R}^3$, is defined as the 3D motion field describing the motion of every visible 3D point between two time steps:

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \mathbf{v}\left(\mathbf{x}_t\right). \tag{2.3}$$

The 3D motion of every visible scene point induces a motion in the image domain, which is defined as *image flow* and corresponds to the 2D motion field $\mathbf{u}(\mathbf{x}) : \mathbf{\Omega} \rightarrow \mathbb{R}^2$, given by the projection of the scene flow. Figure 2.3 shows the motion and projection of a 3D point belonging to a rigid surface. An illustration of 2D and 3D motion fields is present in Figure 3.1. Note that the image flow should not be confused with the *optical flow*, which is defined as the apparent motion of the brightness patterns in the image, since they both may not coincide. For instance, not every 3D motion in the scene yields to a change in the brightness images, e.g., the motion of an untextured surface, and conversely, not every change in brightness is brought by a the motion of a 3D surface, e.g., a change in illumination.

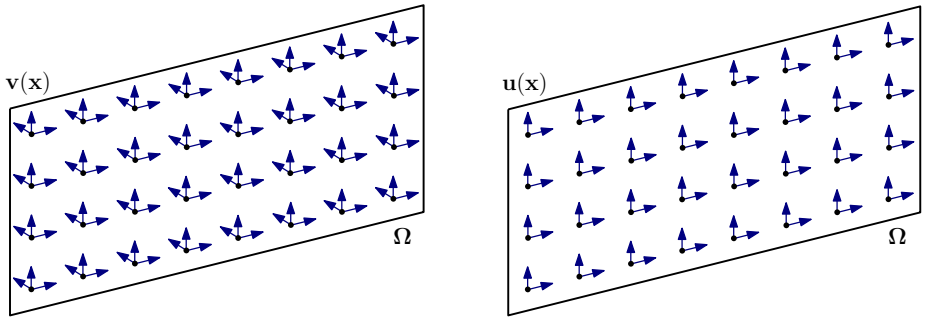


Fig. 2.4: *From left to right: (a) 3D and (b) 2D motion fields represented in the image domain.*

2.2 Rigid body motion

On a *rigid surface* the distance between any two given points remains constant in time regardless of external forces exerted on it. Accordingly, a *rigid body motion*, or rigid motion, is defined as a motion which preserves distance between points of the moving surface. The group action of a rigid body transformation can be written as $T(\mathbf{X}) = \mathbf{R}\mathbf{X} + \mathbf{t}$, where $\mathbf{t} \in \mathbb{R}^3$ is a translation, and $\mathbf{R}$ is a 3×3 matrix belonging to $SO(3)$, the *special orthogonal* group of rotation matrices in $\mathbb{R}^3$, defined as:

$$SO(3) = \left\{ \mathbf{R} \in \mathbb{R}^{3\times 3} : \mathbf{R}\mathbf{R}^T = \mathbf{I}_{3\times 3},\, \det(\mathbf{R}) = \pm 1 \right\}. \tag{2.4}$$

A rigid body motion can be seen a rotation $\mathbf{R}$ followed by a translation $\mathbf{t}$, as is illustrated in Figure 2.5. Using homogeneous coordinates, under the action of rigid motion $\mathbf{G}$, a 3D point $\tilde{\mathbf{X}} = (\mathbf{X}, 1)^T$ is transformed into $\tilde{\mathbf{X}}'$, according to:

$$\tilde{\mathbf{X}}' = \mathbf{G}\tilde{\mathbf{X}}, \quad \text{with} \quad \mathbf{G} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} \in SE(3), \tag{2.5}$$

with $SE(3)$ the *special Euclidean* group representing rigid transformations in $\mathbb{R}^3$:

$$SE(3) = \left\{ (\mathbf{R}, \mathbf{t}) : \mathbf{t} \in \mathbb{R}^3, \mathbf{R} \in SO(3) \right\} = \mathbb{R} \times SO(3). \tag{2.6}$$

The group of matrices, SE(3), is a group with the multiplication as *operation*, and the identity matrix, $\mathbf{I}_{4\times 4}$, as the *identity*. Therefore, the composition of two rigid motion, $\mathbf{G}_1$ and $\mathbf{G}_2$, is done by a matrix multiplication, yielding to the rigid motion $\mathbf{G} = \mathbf{G}_1\mathbf{G}_2$. Moreover, every rigid motion $\mathbf{G}$ has inverse an inverse given by $\mathbf{G}^{-1}$. It is important to remark that $SE(3)$ and $SO(3)$ are *Lie groups*, i.e., groups with a smooth manifold structure. Any Lie group $\mathbb{G}$ defines a *Lie algebra* $\mathfrak{g}$ through an exponential function. Below we define the exponential function relating the Lie group $SE(3)$ with its Lie algebra $se(3)$. Unlike Lie groups, Lie algebras are linear spaces allowing to understand and solve problems on the Lie group in a easier way.

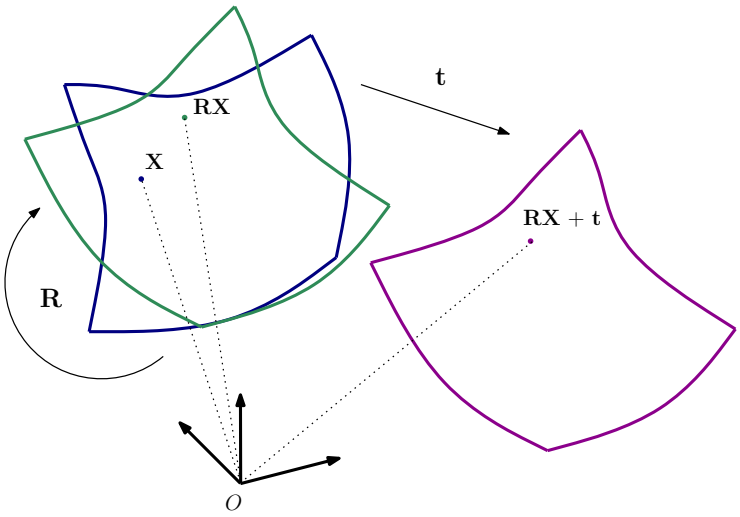


Fig. 2.5: *Rigid body motion represented as a rotation followed by a rotation.*

2.2.1 Twist representation

Since any $\mathbf{G} \in SE(3)$ has only 6 degrees of freedom, a more convenient and compact representation is the 6-parameter twist. Every rigid motion can be described as a rotation around a 3D axis $\omega = (\omega_X, \omega_Y, \omega_Z)^T$ and a translation along this axis, giving as a function of a 3D translation $\tau = (\tau_x, \tau_y, \tau_z)^T$, see Figure 2.6. Therefore it can be shown that for any arbitrary $\mathbf{G} \in SE(3)$ there exists an equivalent $\xi \in \mathbb{R}^6$ twist representation. A twist $\xi = (\omega, \tau)$ can be converted into the $\mathbf{G}$ representation with the following *exponential map*:

$$\mathbf{G} = e^{\hat{\xi}} = \mathbf{I}_{4\times 4} + \hat{\xi} + \frac{(\hat{\xi})^2}{2!} + \frac{(\hat{\xi})^3}{3!} + \cdots , \tag{2.7}$$

where

$$\hat{\xi} = \begin{pmatrix} \hat{\omega} & \tau \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix}, \quad \text{with} \quad \hat{\omega} = \begin{pmatrix} 0 & -\omega_Z & \omega_Y \\ \omega_Z & 0 & -\omega_X \\ -\omega_Y & \omega_X & 0 \end{pmatrix}. \tag{2.8}$$

Every twist vector ξ corresponds to a 4×4 matrix $\hat{\xi} \in se(3)$, which is the Lie algebra of $SE(3)$, via the *hat operator* given by the left side of equation (2.8). Similarly, every rotation vector ω is related to a 3×3 skew-symmetric matrix $\hat{\omega} \in so(3)$, the Lie algebra of $SO(3)$, via the right side of equation (2.8). Alternatively, the exponential map (2.7) can be expressed in a compact form as follows:

$$e^{\hat{\xi}} = \begin{pmatrix} e^{\hat{\omega}} & A\tau \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix}, \tag{2.9}$$

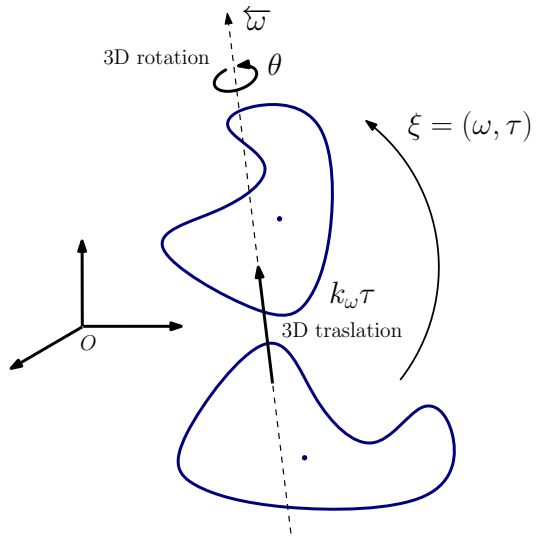


Fig. 2.6: *Motion induced by twist $\xi = (\omega, \tau)$. The 3D rotational component is expressed as $\omega = \theta\overleftarrow{\omega}$, where scalar $\theta = |\omega|$ is the amount of rotation, and the unitary vector denoted by the over-arrow, $\overleftarrow{\omega} = \omega/|\omega|$, is the axis of rotation. The translational component, τ, gives the motion along the 3D axis by $k_\omega \tau$, where $k_\omega = \hat{\omega}/|\omega|^2$. The 3D axis is a line in $\overleftarrow{\omega}$ direction going through the point $\hat{\omega}\tau/|\omega|^2$.*

where the rotation matrix $e^{\hat{\omega}}$ is given by Rodrigues's formula:

$$e^{\hat{\omega}} = \mathbf{I}_{3\times 3} + \frac{\hat{\omega}}{|\omega|}\sin|\omega| + \frac{\hat{\omega}^2}{|\omega|^2}(1 - \cos|\omega|), \tag{2.10}$$

and

$$A = \mathbf{I}_{3\times 3} + \frac{\hat{\omega}}{|\omega|^2}(1 - \cos|\omega|) + \frac{\hat{\omega}^2}{|\omega|^3}(|\omega| - \sin|\omega|), \quad \text{for } \omega \neq \mathbf{0}. \tag{2.11}$$

The exponential map (2.7), or equivalently (2.10), goes from $se(3)$ to $SE(3)$. Correspondingly, for each $\mathbf{G} \in SE(3)$ there exists a twist representation given by the *logarithmic map*:

$$\xi = \log(\mathbf{R}, \mathbf{t}) = (\omega, A^{-1}\mathbf{t}), \tag{2.12}$$

where

$$\hat{\omega} = \frac{1}{2\sin\theta}(\mathbf{R} - \mathbf{R}^T), \quad \text{with} \quad \theta = \cos^{-1}\left(\frac{\text{trace}(\mathbf{R}) - \mathbf{1}}{2}\right), \quad \text{for } \mathbf{R} \neq \mathbf{I}_{3\times 3}, \tag{2.13}$$

and

$$A^{-1} = \mathbf{I}_{3\times 3} - \frac{\hat{\omega}}{2} + \frac{2\sin|\omega| - |\omega|(1 + \cos|\omega|)}{2|\omega|^2\sin|\omega|}\hat{\omega}^2, \quad \text{for } \omega \neq \mathbf{0}. \tag{2.14}$$

2.2.2 Small-rotation model

Any given rotation vector ω encodes the amount, or angle, of rotation in its magnitude, $|\omega|$, and the axis of rotation as the unitary vector $\omega/|\omega|$. In the case of the angle of rotation approaches zero, Rodrigues's formula (2.10) yields to:

$$\lim_{|\omega|\to 0} e^{\hat{\omega}} = \mathbf{I}_{3\times 3} + \hat{\omega}\,. \tag{2.15}$$

Therefore, for a small-rotation rigid motion, the rotation matrix is well approximated as $e^{\hat{\omega}} \approx \mathbf{I}_{3\times 3} + \hat{\omega}$ (Fang and Huang, 1983). Similarly, if $|\omega| \to 0$ we have $A \approx \mathbf{I}_{3\times 3}$ in equation (2.11) and the full rigid motion can be approximated by:

$$e^{\hat{\xi}} \approx \mathbf{I}_{4\times 4} + \hat{\xi}, \tag{2.16}$$

which implies that $\mathbf{t} \approx \tau$. This linear approximation is useful to analyze a small rigid motion in the 3D space and in the image domain. Moreover, equation (2.16) allows to write the scene flow induced by a small-rotation rigid motion as a linear function of the twist. Let ξ be a twist motion for a small rotation, under the rigid motion, the 3D point $\mathbf{X}$ is transformed into $\mathbf{X}'$ inducing the scene flow given by:

$$\mathbf{X}' - \mathbf{X} = ([\,\mathbf{X}]_\times \,|\, \mathbf{I}_{3\times 3}\,)\,\xi, \tag{2.17}$$

where the operator $[\cdot]_\times$ takes the elements of $\mathbf{X}$ to construct a skew-symmetric matrix (the cross product matrix), as is done in the right side of equation (2.8).

2.3 Total variation

Total variation was introduce to Computer Vision by Rudin, Osher and Fatemi (ROF) in their seminal work (Rudin *et al.*, 1992) for image denoising. The ROF denoising model can be written as:

$$\min_u \int_{\mathbf{\Omega}} |\nabla u(\mathbf{x})| d\mathbf{x} + \frac{\eta}{2} \|u - f\|_2^2\,, \tag{2.18}$$

where $\mathbf{\Omega} \subset \mathbb{R}^2$ denotes the image domain, and $f : \mathbf{\Omega} \to \mathbb{R}$, the degraded observed image. The right side of the model (2.18) is the *fidelity term*, which encourages the restored image u to be close to the input image f in L_2, the space of square-integrable functions. The left term is the *total variation* (TV), which is used a Tikhonov regularization and has the capability to preserve discontinuities (edges) while removing noise and other undesired details, see Figure 2.7. The constant η balances the tradeoff between a good fit to the data and a smooth solution. In the TV term, $\nabla u(\mathbf{x})$ is the gradient of u evaluated at $\mathbf{x}$ and its magnitude is given by:

$$|\nabla u(\mathbf{x})| = \sqrt{\frac{\partial u}{\partial x}^2(\mathbf{x}) + \frac{\partial u}{\partial y}^2(\mathbf{x})}\,. \tag{2.19}$$

Fig. 2.7: *ROF-model denoising. From left to right: (a) original image, (b) noisy image and (c) denoised image. (Images courtesy of Mark Werlberger).*

The solution of (2.18), which is also known as the *primal formulation*, is a challenge, since is non-differentiable at $|\nabla u(\mathbf{x})|$ and is highly nonlinear. For this reason the *dual formulation* of the ROF-model has also been studied (Chan *et al.*, 1999) and has been shown that the desired denoised image u^* can be solved by:

$$u^* = f + \frac{1}{\eta}\nabla \cdot \mathbf{w}^*, \tag{2.20}$$

where $\mathbf{w}^* : \mathbf{\Omega} \to \mathbb{R}^2$ is the solution of the dual problem:

$$\min_{|u|\leq 1} \|\nabla \cdot \mathbf{w} + \lambda f\|_2^2 \,. \tag{2.21}$$

The solution of the dual formulation presents its own challenges since it brings extra constraints and its solution is not unique, see (Zhu *et al.*, 2010) for more details. However, the dual problem was the first to allow an exact solution of the ROF model by using a projection algorithm (Chambolle, 2004). The ROF model (2.18) has been successfully used in other computer vision tasks. Particularly, for motion estimation, TV is preferred to other strategies, since TV: *i*) does not penalize discontinuities of the motion, *ii*) does not penalize smooth functions and *iii*) preserves exactly the location of discontinuous edges, see Figure 2.8.

2.3.1 Discrete ROF model

The discrete total variation of a scalar function $u : \mathbf{\Omega} \to \mathbb{R}$ is defined by:

$$\mathbf{TV}(u) = \sum_{\mathbf{x}\in\mathbf{\Omega}} |\nabla u(\mathbf{x})| = \sum_{\mathbf{x}\in\mathbf{\Omega}} \sqrt{u_x^2(\mathbf{x}) + u_y^2(\mathbf{x})}, \tag{2.22}$$

where the components of the gradient $\nabla u = (u_x, u_y)$, are computed using forward differences, see Chambolle (2004). Accordingly, the discrete ROF model is given by:

$$\min_u \;\; \mathbf{TV}_S(u) + \frac{\eta}{2}\sum_{\mathbf{x}\in\mathbf{\Omega}} |u(\mathbf{x}) - f(\mathbf{x})|^2, \tag{2.23}$$

Fig. 2.8: *Comparison of regularization strategies using the Army optical flow dataset by Baker et al. (2007). From left to right: (a) ground truth of the motion, (b) motion estimate using l_2-regularization (Horn and Schunck, 1981), and (c) motion estimate using TV-regularization (Wedel et al., 2009). Unlike l_2-regularization that over-smooths the solution because its quadratic penalty, TV respects motion discontinuities thanks to its l_1-based penalty. (Images courtesy of Mark Werlberger)*

Chambolle's projection algorithm

Problem (2.23) can be solved using the projection algorithm by Chambolle (2005), which works on the dual formulation and provides the solution:

$$u(\mathbf{x}) = f(\mathbf{x}) + \frac{1}{\eta}\nabla \cdot \mathbf{p}(\mathbf{x}) \tag{2.24}$$

where the dual variable $\mathbf{p} : \mathbf{\Omega} \rightarrow \mathbb{R}^2$ is computed iteratively as:

$$\mathbf{p}^{n+1}(\mathbf{x}) = \frac{\mathbf{p}^n(\mathbf{x}) + (\eta\nu)\nabla u^n(\mathbf{x})}{\max\left\{1, |\mathbf{p}^n(\mathbf{x}) + (\eta\nu)\nabla u^n(\mathbf{x})|\right\}} \tag{2.25}$$

with $\mathbf{p}^0 = \mathbf{0}$ and $\nu < 1/4$. However, this approach is only a choice between a large set of alternatives to solve this kind of problems, see Chambolle and Pock (2011).

Arrow-Hurwicz primal-dual algorithm

Algorithms that simultaneously solve the primal and dual formulation converge significantly faster than methods based solely on the primal or dual formulation. An efficient and simple primal-dual formulation is presented by Zhu and Chan (2008), where the dual variable is iteratively computed as:

$$\mathbf{p}^{n+1}(\mathbf{x}) = \frac{\mathbf{p}^n(\mathbf{x}) + (\nu_1)\nabla u^n(\mathbf{x})}{\max\left\{1, |\mathbf{p}^n(\mathbf{x}) + (\nu_1)\nabla u^n(\mathbf{x})|\right\}} \tag{2.26}$$

and the solution u is iteratively given by:

$$u^{n+1}(\mathbf{x}) = \frac{u^n(\mathbf{x}) - \nu_2\nabla \cdot \mathbf{p}(\mathbf{x}) + (\eta\nu_2)f}{1 + \eta\nu_2} \tag{2.27}$$

with $\mathbf{p}^0 = \mathbf{0}$ and $u^0 = \mathbf{0}$, and the parameters $\nu_1 > 0$, $\nu_2 > 0$ and $\nu_1\nu_2 < 1/2$.

Staircaising Effect

When total variation is used to regularize, it can suffer from the staircaising problem. This effect is characterized by artificial constant areas in the solution. It is possible to correct these artifacts by replacing the l_1 norm in the TV term (2.22) by the Huber norm (Werlberger *et al.*, 2009):

$$|u|_\alpha = \begin{cases} \frac{|u|^2}{2\alpha} & |u| \leq \alpha \\ |u| - \frac{\alpha}{2} & |u| > \alpha \end{cases}, \tag{2.28}$$

where $\alpha > 0$ is a small parameter balancing between the l_1 and the l_2 penalization. Quadratic regularization is applied on small values while TV regularization is used with larger values, thus diminishing the staircase effect. The Huber norm can be easily used with any of the algorithms presented above.

The starcaising problem can also be solved by penalizing the variation of higher derivatives of u, as is propesed by Bredies *et al.* (2010) with the *total generalized variation* (TGV). Particularly, the second order TGV is defined by:

$$\mathbf{TGV}^2(u) = \min_v \ \alpha_0 \sum_{\mathbf{x}\in\mathbf{\Omega}} |\nabla u(\mathbf{x}) - v(\mathbf{x})| + \alpha_1 \sum_{\mathbf{x}\in\mathbf{\Omega}} |\nabla v(\mathbf{x})|, \tag{2.29}$$

where α_0 and α_1 balance between the first and second order derivative terms. Penalizing the variation of the second order derivative benefits smooth solutions, eliminating the staircase effect.

2.3.2 Vectorial total variation

The idea of total variation can be extended to a vector-valued function $\mathbf{u} : \mathbf{\Omega} \to \mathbb{R}^N$. Goldluecke *et al.* (2012) analyze some of the options to define a *vectorial total variation* (VTV). Such approaches can be divided in two classes: those that compute the TV channel by channel and those that consider the Riemann geometry of the image manifold. An intuitive way to define the VTV is to sum up the contribution of every channel. The channel-by-channel VTV is defined by:

$$\mathbf{TV}_S(\mathbf{u}) = \sum_{i=1}^{N} \mathbf{TV}(u_i). \tag{2.30}$$

This definition allows a fast and simple regularization separately by channel, where any of the two algorithms shown above can be used. However, since there is no coupling between channels, the correlation between the different components is not exploited. For instance, depending of the nature of $\mathbf{u}$, its discontinuities may appear at the same position in multiples channels and this information can benefit the regularization to preserve the multidimensional edge. This is the case when TV regularization is applied to color images or a motion field, the latter is the main concern of this thesis.

By considering $\mathbf{u}$ as a two-dimensional Riemann manifold, the *derivative matrix*, $\mathrm{D}\mathbf{u} := (\nabla u_1, \nabla u_2 ..., \nabla u_N)^T : \boldsymbol{\Omega} \rightarrow \mathbb{R}^{N\times 2}$, can be used to compute the metric tensor $g(\mathbf{u}) = (\mathrm{D}\mathbf{u})^T \mathrm{D}\mathbf{u}$. In this case, the eigenvector λ_+ corresponding to the largest eigenvalue, gives the direction of the vectorial edge. Goldluecke and Cremers (2010) propose the use of the *Jacobian* J_1 from geometric measure theory, which corresponds to $\sqrt{\lambda}_+$ and can be computed as the largest singular value of matrix $\mathrm{D}\mathbf{u}$. Accordingly, the VTV can be defined by:

$$\mathbf{TV}_{\sigma_1}(\mathbf{u}) = \sum_{\mathbf{x}\in\boldsymbol{\Omega}} \sigma_1(\mathrm{D}\mathbf{u}(\mathbf{x}))\,, \tag{2.31}$$

where σ_1 is the largest singular value of the derivative matrix. The metric $\mathbf{TV}_{\sigma_1}$ has two main properties: it weighs the contribution of every channel, unlike $\mathbf{TV}_S$, and it supports a common edge direction for all channels. This latter property is the main advantage of $\mathbf{TV}_{\sigma_1}$ with respect other metrics. For example, the Frobenius norm of $\mathrm{D}\mathbf{u}$ can be used to define the metric:

$$\mathbf{TV}_F(\mathbf{u}) = \sum_{\mathbf{x}\in\boldsymbol{\Omega}} \|\mathrm{D}\mathbf{u}(\mathbf{x})\|\,, \tag{2.32}$$

allowing a coupling between all channels, but since it does not support a common edge, the vectorial edge is strongly penalized. It can be shown that $\mathbf{TV}_{\sigma_1}(\mathbf{u}) \leq \mathbf{TV}_F(\mathbf{u}) \leq \mathbf{TV}_S(\mathbf{u})$ (Goldluecke and Cremers, 2010), reflecting the softer penalization performed by $\mathbf{TV}_{\sigma_1}$ which protects the vectorial edge.

2.4 Robust Lucas-Kanade framework

The goal of Lucas-Kanade algorithm is to align a *template image*, $T(\mathbf{x})$, to an *input image*, $I(\mathbf{x})$, therefore, it allows to compute optical flow or to track an image patch between two frames. Following (Baker and Matthews, 2004), this problem can be stated as finding the parameter vector $\boldsymbol{\theta} = (\theta_1, ..., \theta_n)^T$ which minimizes a squared sum of differences between the template T an the current image I:

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \sum_{\mathbf{x}} \left[I\left(\mathbf{W}(\mathbf{x}, \boldsymbol{\theta})\right) - T\left(\mathbf{x}\right)\right]^2, \tag{2.33}$$

where the warp $\mathbf{W}(\mathbf{x}, \boldsymbol{\theta})$ maps each template pixel $\mathbf{x}$ to a pixel on the image. The warping function can be defined using an image motion model, *e.g.*, one of those presented in Chapter 4. This approach was introduced by Lucas and Kanade (1981) for image registration, assuming that the optical flow corresponds to a single (continuously varying) motion locally; that is, the variation inside the patch is assumed to be Gaussian. Formulation (2.33) is a least-squares problem, which can be seen as a particular case of *maximum a posteriori* estimation as is shown below. Bayesian reasoning can provide an alternative solution.

2.4.1 Bayesian derivation of the Lucas-Kanade framework

Assume that we want to estimate parameters $\boldsymbol{\theta} = \{\theta_1, ..., \theta_n\}$ given a set of observations $\boldsymbol{\rho} = \{\rho(\mathbf{x}_1, \boldsymbol{\theta}), ..., \rho(\mathbf{x}_L, \boldsymbol{\theta})\}$, where each observation is defined as the residual between the warped image and the template according to $\rho(\mathbf{x}_i, \boldsymbol{\theta}) = I\left(\mathbf{W}(\mathbf{x}_i, \boldsymbol{\theta})\right) - T\left(\mathbf{x}_i\right)$. Let $P(\boldsymbol{\theta})$ be the *a priori* distribution of the parameters. Baye's law converts the prior belief about parameters $\boldsymbol{\theta}$, into *a posterior probability* $P(\boldsymbol{\theta}|\boldsymbol{\rho})$ by using the *likelihood function* as follows:

$$P(\boldsymbol{\theta}|\boldsymbol{\rho}) = \frac{P(\boldsymbol{\rho}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\boldsymbol{\rho})}, \tag{2.34}$$

thus, the *maximum a posteriori* (MAP) estimate is defined as:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \; P(\boldsymbol{\theta}|\boldsymbol{\rho}). \tag{2.35}$$

Because of noise, residuals are not zero, but follow a particular distribution $P(\boldsymbol{\rho}|\boldsymbol{\theta})$. Assuming that the noise of all observations is independent and identically distributed, and since $P(\boldsymbol{\rho})$ does not depend on $\boldsymbol{\theta}$, the MAP estimate becomes:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \; \prod_{\mathbf{x}} P(\rho(\mathbf{x}, \boldsymbol{\theta})|\boldsymbol{\theta})P(\boldsymbol{\theta}). \tag{2.36}$$

The prior $P(\boldsymbol{\theta})$ models the prior distribution of the parameter. In the Lucas-Kanade framework this belief corresponds to the prior distribution of image motions, which is assumed uniform, so that every possible parameter, or motion, is equiprobable and $P(\boldsymbol{\theta})$ can be dropped from (2.36). Therefore, the MAP estimate is equivalent to minimizing the negative log of the likelihood function:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \; -\sum_{\mathbf{x}} \log\left(P(\rho(\mathbf{x}, \boldsymbol{\theta}))\right). \tag{2.37}$$

If the noise is Gaussian, e.g., $P(\rho(\mathbf{x}, \boldsymbol{\theta})|\boldsymbol{\theta}) \sim \mathcal{N}(\rho(\mathbf{x}, \boldsymbol{\theta}), 0, \sigma)$, the MAP estimate applied to the Lucas-Kanade framework, gives:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \; \frac{1}{2\sigma^2} \sum_{\mathbf{x}} \rho^2(\mathbf{x}, \boldsymbol{\theta}), \tag{2.38}$$

where the constant related to σ can be removed.

2.4.2 Robust least-squares

If the noise distribution is not Gaussian, the least-squares estimate may be skewed from the real (or desirable) solution. This is a very common case for motion estimation using the Lucas-Kanade framework. The Gaussian assumption is commonly violated due to occlusion or multiple motions, yielding undesired solutions. Particularly, in a least-squares approach, outliers contribute too much

to the overall solution, because they are not expected in the prior distribution. In order to improve robustness, the estimator should be able to recover the structure that best fits the majority of the data while identifying and rejecting outliers.

The goal of a robust estimator is to find the values for the parameters, $\boldsymbol{\theta}$, that best fit model $\mathbf{W}(\mathbf{x}_i, \boldsymbol{\theta})$ for the observed measurements $\boldsymbol{\rho}$, in cases where the data differs statistically from the model assumptions. For this reason the quadratic, or l^2-, penalty, brought by the Gaussian assumption, is replaced by a function ψ, which is called a *M-estimator* since it is related to the maximum-likelihood estimate, shown in Equation 2.37. Robust least-squares estimation is formulated as:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{\mathbf{x}} \psi\left(\rho^2(\mathbf{x}, \boldsymbol{\theta})\right), \tag{2.39}$$

The specific M-estimator, $\psi(s^2)$, characterizes the insensitivity to outliers or deviations (*robustness*) of the estimation. The case $\psi(s^2) = s^2$ corresponds to the quadratic penalty, where all square residues are treated the same and outliers may skew the solution. If Equation (2.39) is differentiable, the derivative of the M-estimator weighs every observation in the solution, and can be used to analyse the influence of outliers in the estimate. For instance, the *Lorentzian* is a M-estimator defined as:

$$\psi_a(s^2) = \log\left(1 + \frac{s^2}{2a^2}\right), \quad \text{with} \quad \psi'_a(s^2) = \frac{1}{s^2 + 2a^2}, \tag{2.40}$$

has a derivative which decreases as s^2 moves away from zero, reducing the influence of outliers. A less robust M-estimator, i.e., with a slower decreasing derivative, is the *Charbonnier* penalty, is given by:

$$\psi_\varepsilon(s^2) = \sqrt{s^2 + \varepsilon^2}, \quad \text{with} \quad \psi'_\varepsilon(s^2) = \frac{0.5}{\sqrt{s^2 + \varepsilon^2}}, \tag{2.41}$$

corresponding to a differentiable approximation of the l_1 norm. This penalty has shown to perform better than other M-estimators for optical flow estimation (Sun *et al.*, 2010). Figure 2.9 shows these three penalties and their derivatives.

2.4.3 Iterative solution

The robust Lukas-Kanade estimate is defined by:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{\mathbf{x}} \psi\left(\left[I\left(\mathbf{W}(\mathbf{x}, \mathbf{p})\right) - T\left(\mathbf{x}\right)\right]^2\right), \tag{2.42}$$

which is a nonlinear optimization problem, which can be solved by iterative reweighted least squares (IRLS) with the Gauss-Newton algorithm. Assuming that an initial estimate $\boldsymbol{\theta}$ is known, the input image I can be approximated around it, by using the first-order Taylor expansion. Accordingly, (2.42) becomes the problem

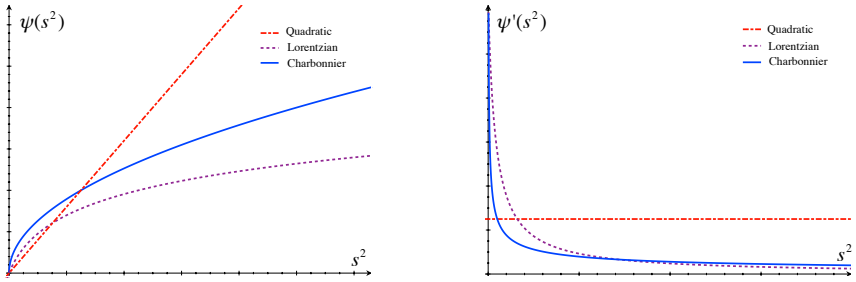


Fig. 2.9: *Comparison of quadratic, Lorentzian ($a = 1$) and Charbonnier ($\varepsilon = 0.001$) penalties.*

of finding an incremental $\Delta\boldsymbol{\theta}$ that minimizes

$$\sum_{\mathbf{x}} \psi \left(\left[\rho(\mathbf{x}, \boldsymbol{\theta}) + I_{\mathbf{x}} \frac{\partial \mathbf{W}}{\partial \boldsymbol{\theta}} \Delta\boldsymbol{\theta} \right]^2 \right), \tag{2.43}$$

with $\rho(\mathbf{x}, \boldsymbol{\theta}) = I\left(\mathbf{W}(\mathbf{x}, \boldsymbol{\theta})\right) - T(\mathbf{x})$ the image residue, and where $I_{\mathbf{x}} = (\partial I/\partial x, \partial I/\partial y)$ denotes the input image gradient and $\partial \mathbf{W}/\partial \boldsymbol{\theta}$ the Jacobian of the warp, at evaluated at $\mathbf{W}(\mathbf{x}, \boldsymbol{\theta})$. Finding the minimum of expression (2.43) requires an iterative solution. Taking the partial derivative with respect to $\Delta\boldsymbol{\theta}$ gives:

$$2 \sum_{\mathbf{x}} \psi'(\rho^2(\mathbf{x}, \boldsymbol{\theta} + \Delta\boldsymbol{\theta})) \left(I_{\mathbf{x}} \frac{\partial \mathbf{W}}{\partial \boldsymbol{\theta}} \right)^T \left(\rho(\mathbf{x}, \boldsymbol{\theta}) + I_{\mathbf{x}} \frac{\partial \mathbf{W}}{\partial \boldsymbol{\theta}} \Delta\boldsymbol{\theta} \right), \tag{2.44}$$

and at the minimum of (2.43), equation (2.44) is zero, so that the incremental scene flow $\Delta\boldsymbol{\theta}$ can be computed by the following iterative procedure:

1. *Initialize*: $\Delta\boldsymbol{\theta} = 0$.
2. Update weights with the current estimation, θ, and compute the increment:

$$\Delta\theta = -\mathbf{H}^{-1} \sum_{\mathbf{x}} \psi'(\rho^2(\mathbf{x}, \boldsymbol{\theta})) \left(I_{\mathbf{x}} \frac{\partial \mathbf{W}}{\partial \boldsymbol{\theta}} \right)^T \rho(\mathbf{x}, \boldsymbol{\theta}), \tag{2.45}$$

   where $\mathbf{H}$ is the Gauss-Newton approximation of the Hessian matrix:

$$\mathbf{H} = \sum_{\mathbf{x}} \psi'(\rho^2(\mathbf{x}, \boldsymbol{\theta})) \left(I_{\mathbf{x}} \frac{\partial \mathbf{W}}{\partial \boldsymbol{\theta}} \right)^T \left(I_{\mathbf{x}} \frac{\partial \mathbf{W}}{\partial \boldsymbol{\theta}} \right). \tag{2.46}$$

3. *Updating step*: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} \circ \Delta\boldsymbol{\theta}$, with "$\circ$" the composition operation.
4. If *termination criteria* stop, otherwise goto 1.

3 Problem Analysis

3.1 Problem definition

The 3D motion field of a scene, or scene flow, is useful for several computer vision applications, such as action recognition, interaction, or 3D modeling of nonrigid objects. Scene flow methods can be classified into two main groups depending on if the 3D structure of the scene is estimated or provided. The first group of methods uses a stereo or multi-view camera system, where both scene flow and depth are estimated from images. The second group uses RGBD images as input. In this case, the depth given by the sensor may be used directly for scene flow estimation. This thesis proposes a new formulation for this second group.

The two main questions for scene flow estimation from RGBD images are: (a) how to fully exploit both sources of data, and (b) which motion model should be used to compute a confident scene flow. Regarding the first question, 2D motion estimation has made a great progress in the last years and powerful tools are available to estimate optical flow from color images (Sun *et al.*, 2010). Optical flow is defined as the apparent motion of brightness pattern in the image and is closely related to image flow, which is the projection of the 3D motion field on the image. However, optical and image flow may not coincide, as is show in Figure 3.1. Moreover, changes on the motion field are not always reflected in the optical flow or can be hard to estimate, such as the motion of untextured surfaces. Also, changes in depth and occlusions are a challenge for current optical flow methods since there is no 3D knowledge to disambiguate the possible 2D motions. Thanks to the RGBD sensor, information about the 3D structure of the scene is available. By using depth data alone it is possible to estimate a global rigid motion, e.g., the camera motion, as is proposed by Newcombe *et al.* (2011) using *Iterative Point Cloud* (ICP) (Besl and McKay, 1992). However, solving for a nonrigid motion field is a more challenging problem having a wider scope. Although it is possible to perform nonrigid registration of depth data (Li *et al.*, 2009), low-noise and high-resolution depth data is required, which prevents the use of current RGBD sensors. The availability of RGBD data opens the door to a new scenario for motion estimation,

where some of these drawbacks can be overcome.

When both color and depth are provided, there is no consensus on how to most effectively combine these two sources of information. A straightforward approach would be to compute the optical flow from RGB images and to infer the scene flow from the depth data. Alternatively, the texture of depth images can be used to help the 2D motion estimation and simultaneously used to compute the changes in depth. It is also possible to generate colored 3D point clouds or 3D patches by exploiting the depth data and try to estimate a 3D transformation consistent with the input images. Nevertheless, it is not necessary to explicitly represent points in 3D: the scene structure can be represented in the image domain, where color and depth data can be coupled using a projective function. This way, depth changes influence the motion in the image domain and consistency constraints can be formulated jointly over the color and depth images. We survey some of these alternatives by reviewing the related work to this thesis in the next Section.

The second question, concerning the representation of the motion, has been addressed less. Scene flow estimation using intensity and depth is an ill-posed problem and regularization is needed. Instead of solving for 2D motions, as in optical flow estimation, the 3D motion vector of each point can be directly solved to minimize color and intensity constraints in a *data term*. Moreover, the whole 3D motion field can be regularized to get spatially smooth solutions while preserving discontinuities. Since depth data is available, it can be used in the regularization to preserve motion discontinuities along depth edges, where independent motions are probable to appear, and also can be exploited to define a prior of the local rigidity. However, solving for a piecewise smooth solution of the 3D motion field may not be the best choice for some motions of interest. For example, a 3D rotation of a rigid surface induces a great variety of 3D motions that are hardly well regularized by such an approach. A similar issue occurs when the RGBD sensor is moving with respect to the scene. For this reason the motion representation should be able to model rigid and nonrigid components of the scene flow, while favoring the regularization procedure. Moreover, scene flow estimation can be formulated in 3D space or in the image domain and is not clear in which domain the parametrization of the scene yields more accurate results. While 3D space is the natural domain of the scene flow, RGBD images are sensed and provided in the image domain and this 2D parametrization can be exploited.

This thesis aims to make a step forward towards a better understanding of scene flow estimation from RGBD images. For this reason, in order to cover the two questions previously formulated, this study focuses in four main aspects: *i*) the coupling of intensity and depth, *ii*) the representation of motion, *iii*) the constraints of motion in RGBD data and *iv*) the regularization. Temporal consistency of the scene flow is out of the scope of this thesis and the estimation problem is formulated on a frame-to-frame scenario.

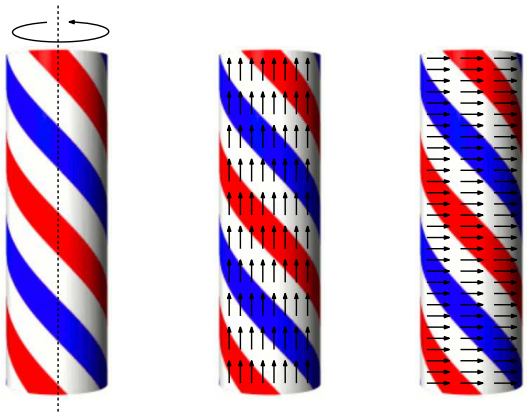


Fig. 3.1: *Difference between apparent and real motion. From left to right: (a) a barber's pole rotating to the right, (b) optical flow and (c) 2D motion field. While the pole is rotating, the color patterns make perceive an upward motion, which does not correspond with the motion field.*

3.2 Related work

Scene flow was first introduced by Vedula *et al.* (1999) as the 3D motion field of the scene. Since this seminal work several approaches have been proposed to compute the 3D motion field. If a stereo or multi-view (SoMV) camera system is available, scene flow can be computed by enforcing consistency with the observed optical flows (Vedula *et al.*, 2005), by a decoupled (Wedel *et al.*, 2008) or joint (Huguet and Devernay, 2007; Basha *et al.*, 2010) estimation of structure and motion, or by assuming a local rigidity of the scene (Vogel *et al.*, 2011). On the other hand, if a depth sensor is available the estimation of the structure is not needed and color and depth information can be simultaneously used. In this case scene flow can be computed by considering the depth data as an additional image channel (Spies *et al.*, 2000), by coupling color and depth in the image domain with a projective function (Quiroga *et al.*, 2012, 2014b) or by reasoning directly in 3D space (Hadfield and Bowden, 2014; Hornacek *et al.*, 2014). Although in this thesis we assume that a depth sensor is available and the estimation of the structure is not needed, below we summarize the most relevant previous methods based on SoMV systems and RGBD images. Scene flow estimations from SoMV and RGBD images are very related, as well as with optical flow estimation, so that their jointly survey could help to a better understanding of this motion estimation problem. A more detailed survey is done for RGBD-based methods in order to clarify the main contributions of this thesis.

3.2.1 Scene flow from stereo or multi-view

Most scene flow methods assume a stereo, or multi-view camera system, in which the motion and the geometry of the scene are jointly estimated, in some cases, under a known scene structure. Inspired by optical flow methods, the most common approach for estimating scene flow is to perform an optimization on a global energy,

including photometric constraints and some regularization. Since optical flow is an approximation of the projection of the 3D motion field on the image plane, an intuitive way to compute scene flow is to reconstruct it from the optical flow measured in a multi-view camera system, as proposed by (Vedula *et al.*, 1999), or including a simultaneously structure estimation as by Zhang and Kambhamettu (2001). Although it is possible to estimate optical dealing with large displacements (Brox and Malik, 2011) and preserving small details (Xu *et al.*, 2012), it is difficult to recover a scene flow compatible with several observed optical flows that may be contradictory. Moreover occlusion handling is a challenge since it could affect both the per-view optical flow computation and the posterior scene flow estimation using multiple views.

Some authors introduce constraints from a fully calibrated stereo or multi-view camera system. Huguet and Devernay (2007) simultaneously compute the optical flow field and two disparities maps by extending the variational optical flow framework by Brox *et al.* (2004) to a stereo system: optical flow constraints for each camera (left, right) and stereo constraints for each time $(t, t+1)$. This way a temporal consistency is enforced on the disparity map estimation between t and $t+1$. Nevertheless, this approach may be impractical due to its high computational time. Wedel *et al.* (2008) decouple disparity and motion estimation to achieve real-time computation, since each problem can be separately solved more efficiently. Although the spatio-temporal information is not fully exploited, the optical flow is enforced to be consistent with the computed disparities. Valgaerts *et al.* (2010) generalize the stereo-based scene flow formulation by embedding epipolar constraints in the energy functional, showing that scene flow and the stereo structure can be simultaneously estimated. Since motion and depth discontinuities do not necessarily coincide, regularization is performed separately. All these methods suffer from the smoothness constraints brought by 2D parametrization, especially due to the regularization on the optical flow rather than directly on the scene flow. Basha *et al.* (2010) improve estimation by formulating the problem as a point cloud in 3D space supporting multiple views. In this method scene flow and the estimated disparity are regularized independently using total variation (TV). In a closely related approach, Vogel *et al.* (2011) regularize the scene flow by encouraging a locally rigid 3D motion field, outperforming TV regularization. In a later work,Vogel *et al.* (2013) present a more constrained assumption and represent the 3D scene by a collection of rigidly moving planes. The scene flow is computed by iteratively alternating three procedures: plane fitting, pixel-to-plane assignation and motion estimation for every plane. This methods achieves state-of-the-art results in the KITTI stereo flow benchmark Geiger *et al.* (2012).

Alternatively, other methods formulate scene flow estimation as a 3D tracking problem. On one hand, it is possible to use an explicit model of the scene and compute a 3D motion consistent with the pre-frame reconstruction and the photometric information. Neumann and Aloimonos (2002) use a subdivision surface model, where the shape and motion of an object are initialized independently, frame to frame, and then refined simultaneously. Fukurawa and Ponce (2008) track an initial 3D mesh model of the scene by a local estimation and a posterior full

deformation of the mesh. The motion of each vertex is estimated using photometric information across different views, afterwards the full mesh is deformed enforcing smoothness and local rigidity. Another possibility is to track a sparse set of 3D points or surface elements (Carceroni and Kutulakos, 2008; Devernay *et al.*, 2006). Carceroni and Kutulakos (2008) model the scene as a set of planar surfels (surface elements) and solve for their pose and motion using photometric constraints across different views. This approach requires a well-controlled lighting and acquisition setup, and because its complexity, scene flow solution is only suitable in a limited volume. Devernay *et al.* (2006) directly get a set of 3D trajectories for 3D points from which the scene flow is derived. However, this method suffers from drift problems and its proposed method for point visibility handling is difficult to perform.

3.2.2 Scene flow from RGBD images

The first work using intensity and depth was by Spies *et al.* (2000), where the optical flow formulation by Horn and Schunck (1981) is extended to include depth data. In this approach, depth data is used simply as an additional channel in the variational formulation of the optical flow trough the definition of a *range flow* equation encouraging the consistency between 2D motion and changes in depth. This way both intensity and depth image gradients are combined to jointly estimate the optical and the range flow, which are enforced to be smooth.

Lukins and Fisher (2005) extend this approach to multiple color channels and one aligned depth image. Nevertheless, in both methods, the scene is assumed to be captured by an orthographic camera and there is no an optimal coupling between optical and range flows. Moreover, due to the Horn-Schunck framework, these methods suffer from the early linearization of the constancy constraints and from over-smoothing along motion boundaries because of the L^2-regularization. The coupling issue can be solved by using a projective function that allows to constrain the 3D motion in the image domain. Because depth information is available it is possible to define a projective function for every pixel using its depth measure. This way changes in depth are used to support the constraint of motion in the image domain. Following this idea, Letouzey *et al.* (2011) project the scene flow in the image domain using a projective matrix to relate 3D motion and image flow. Thus scene flow can be directly solved by using photometric consistency constraints. However, this method uses a projective function that only supports smalls changes in depth, also it suffers for the well known drawbacks of the Horn-Schunck framework. Furthermore the depth data is not fully exploited since there is no range flow constraint and is only used to support the projective function.

Rather than computing a dense scene flow, we define a 2D warping function to couple image motion and 3D motion (Quiroga *et al.*, 2012), allowing for a joint local constraint of scene flow on intensity and depth data. The local 3D motion is modeled as a 3D translation and solved as a tracking problem in RGBD data, using a Lucas-Kanade based framework (Lucas and Kanade, 1981; Baker and Matthews, 2004). Unlike the projective matrix (Letouzey *et al.*, 2011), this warping function

allows to project the true 3D motion on the image and can be approximated by a linear model, to iteratively solve for the scene flow. Although the method is able to deal with large displacements, it fails on untextured regions and more complex motions, such as rotations. In order to solve for dense scene flow, a regularization procedure that preserves motion discontinuities is required. Inspired by optical flow methods, the 3D motion field can be assumed to be piecewise smooth, and total variation (TV) can be applied as regularizer.

The work by Herbst *et al.* (2013) follows this idea, but as (Spies *et al.*, 2000), it lacks a coupling between optical and range flows, and the regularization is done on the optical flow rather than on the scene flow. Such that approach lacks of the accuracy to recover a confident scene flow, and it suffers of the same drawbacks that optical flow methods. The regularization of 3D motion field is more reliable than encouraging smoothness on the apparent motion in the image domain. In (Quiroga *et al.*, 2013) we present a variational extension of the local method for scene flow (Quiroga *et al.*, 2012). In this work, the tracking approach in intensity and depth is regularized to compute a dense scene flow enforcing piecewise smoothness. In order to preserve motion discontinuities along depth edges, a weighted TV is applied on each component of the 3D motion field.

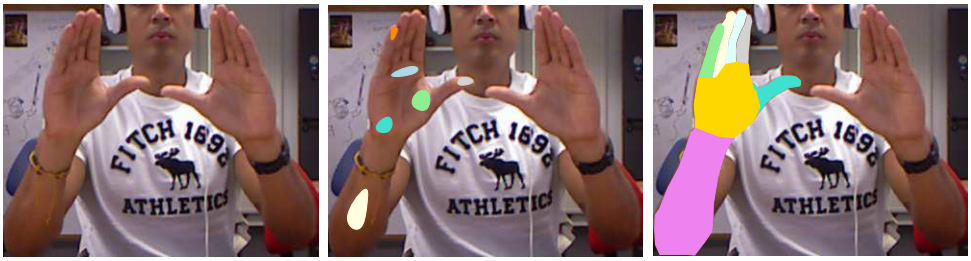
All these methods assume spatial smoothness of the scene flow, which is a reasonable assumption for translational motions but not the optimal choice for rotations. Under a rotation, even close scene points present different 3D motions. Particularly, in the case of a moving camera the regularization of the motion field can be a challenge. For this reason, we present an over-parametrization of the scene flow (Quiroga *et al.*, 2014a), where each scene point is allowed to follow a rigid body motion. This way, the regularization can be done on a field of rigid motions, favoring piecewise solutions, which is a better choice for real scenes. A similar idea is presented by Rosman *et al.* (2012), where a regularization of a field of rigid motions is proposed. However, our work (Quiroga *et al.*, 2014a) differs from such that by Rosman *et al.* (2012), in three ways. First, a more compact representation is used of the rigid-body motion via the 6-parameter twist representation, instead of a $\mathbb{R}^{12}$ embedding. Second, this approach solves and regularizes the rigid motion field at the same time. Finally, it is proposed a decoupled regularization of the rotational and translational fields, which simplifies the optimization and allows the use of different TV strategies on each field. Also, as we present in (Quiroga *et al.*, 2013), a depth-based weighting function is used to avoid penalization along surface discontinuities and a warping function is defined to couple the twist motion and the optical flow. A very similar warp is presented by Kerl *et al.* (2013) to solve for a global rigid motion from RGBD images. However, we use the warping function to locally constrain the rigid motion field in the image domain, while the global rigid motion estimation can be seen as a particular case of this formulation. Moreover, unlike Kerl *et al.* (2013), a depth consistency constraint is defined to fully exploit both sources of data.

An alternative approach is the representation of the scene in 3D space. Hadfield and Bowden (2014) estimate the scene flow using particle filtering in a 3D colored point cloud representation of the scene. In that approach, a large set of 3D

motion hypotheses must be generated and tested for each 3D point, leading to high computational costs. Also, although the propagation of multiple hypotheses could avoid to get stuck at local minima, it is not clear the benefit brought by formulating the problem on an unstructured 3D point cloud. Hornacek *et al.* (2014) compute scene flow by matching rigid patches of 3D points identified as inliers of spheres, where RGB and depth information are simultaneously used to define a matching cost. Rather than solve for 3D motions, this approach estimates an optimal rigid motion for every patch, by generating and propagating a set of hypotheses. The scene flow is Iteratively regularized by enforcing local rigidity, which is done as a labeling optimization problem. This approach is very related to our work (Quiroga *et al.*, 2014a), in the sense that both solve for a dense field of rigid motions and encourage local rigidity. However, these two approaches differ in several aspects, such as the kind of RGBD consistency constraints, and the optimization strategy: while Hornacek *et al.* (2014) use discrete optimization via a modified *PatchMatch* (Hornacek *et al.*, 2013), Quiroga *et al.* (2014a) use a Gauss-Newton algorithm to perform continuos optimization. Nevertheless, the main difference concerns our scene flow framework that minimizes a local/global energy, which enables an adjustable combination between local and piecewise rigidity.

3.3 Proposed approach

In this thesis, we take advantage of the fact that most real scenes can be well modeled as locally or piecewise rigid, i.e., the scene is composed of 3D independently rigid components. We define a *semi-rigid scene* as one that its 3D motion field is well approximated as locally or piece-wise rigid, or a combination of both, see Figure 3.2. This way scene flow estimation can benefit from the rigid properties of the scene while allowing motion discontinuities and outliers, being able to model complex motion patterns. In order to exploit this assumption, we define a framework for scene flow estimation from RGBD images, using an over-parametrization of the motion. We represent the scene flow as a vector field of rigid body motions, which is expressed as a field of twist motions and solved directly in the image domain. This representation helps the regularization process, which, instead of directly penalizing variations in the 3D motion field, encourages piecewise smooth solutions of rigid motions by using a weighted TV regularization to preserve motion discontinuities along surface edges. Moreover, this semi-rigid approach can be constrained in the image domain to fully exploit intensity and depth data. For this purpose, we define a projective function that makes it possible to constrain every rigid motion on the image and to define an incremental solver, which is embedded in a robust least-square estimation to exploit local rigidity. Thus, we can solve for the local twist motion that best explains the observed intensity and depth data, gaining robustness under noise. The local solver, in conjunction with the TV regularization of the twist field, provides an adjustable combination between local and piecewise rigidity. This formulation is flexible enough to support different data constraints and regularization strategies, and it can be adapted to more specialized problems. By using the same general framework, it is possible to model the scene flow as



(a) (b) (c)

Fig. 3.2: *Local and piecewise rigidity assumptions. (a) Input image, (b) local rigidity and (c) piecewise rigidity. Every color patch indicates that inner scene points present (approximated) the same rigid motion. Local rigidity favor the best rigid motion explaining the motion inside every patch. Piece-wise rigidity encourage a set of rigid moving segments.*

a global rigid motion plus a nonrigid residual, which is particularly useful when estimating the motion of deformable objects in conjunction with a moving camera. As we present in the experiments, this framework is flexible enough to estimate a global rigid body motion or to solve for a general 3D motion field in challenging setups. This way, we are able to model the motion of the scene as a global rigid motion and a nonrigid residual.

4

Scene Motion Representation

Real world scenes are composed of a set of 3D surfaces, which in the more general case, are free to move and deform independently. Using a depth sensor we are provided with a finite set of samples of the scene. This is assumed to be enough to represent the most important details of the 3D structure and to observe the motions of interest in the scene. In the RGBD representation, every valid sample of the scene, or *scene point*, is characterized by a RGB color and a depth measure. If between two time steps there exists a motion of any 3D surface or of the RGBD camera, or both simultaneously, the RGBD representation of the scene changes, enabling the estimation of the 3D motion field.

Perhaps the most intuitive approach to solve for the scene flow is to use the RGBD data without imposing any rule about the 3D motion. If any scene point moves freely there are no limitations for the motions that can be modeled. However, this independence between scene points, which may favor the richness of motions modeled, leads to a very ill posed problem for two main reasons. First, the color of every scene element is not discriminative since several scene points may have the same color. Moreover, because of the presence of noise and illumination changes, color-based consistency assumptions are commonly violated. Second, using each depth measure alone does not make it possible to constrain 3D motion since depth measures change with the motions in the scene. Therefore, any scene point with a similar color would be a potential candidate for the new position of the scene point and the 3D structure information cannot be exploited.

An appropriate motion model is required to reliably estimate the 3D motion of every scene point . This model should be general enough to cover a wide variety of interesting 3D motions but also it should allows constraints on the set of motions, making it possible to state the scene flow formulation as a well-posed problem. Most real world scenes are composed of 3D surfaces that can be well approximated as locally or piece-wisely rigid, even very complex scenes. We define a semi-rigid surface as a surface that any of its feasible motions can be well approximated as a locally or piece-wise rigid motion, or a combination of both. This assumption is fundamental because allows the use of a smoothness prior on the motion of scene

points. Note that the assumption of semi-rigidity does not imply planar surfaces. On a locally rigid surface, the 3D distance between scene points remains constant and the 3D motions of these points are closely related, as well as the motion of their projections on the image. Usually this assumption is exploited by assuming local solutions of the optical flow or the 3D motion field. However, points belonging to a rigid surface only have the same 3D motion if the rotational component is zero. Otherwise, every point performs a different 3D displacement, as is observed from Equation (2.17). Although that kind of regularity yields a better statement of the problem, it does not totally exploit the properties of a locally rigid scene. Similarly, the properties of a piece-wise rigid scene explain the benefits of using a TV-based regularization, as is done in optical flow methods, but it is the chosen representation of motion which determines the degree of utilization of the semi-rigid assumption in the motion solution.

In this thesis we go beyond previous approaches by considering a semi-rigid scene. For this purpose, scene flow is represented by a field of twist motions, where local and piece-wise rigidity can be fully exploited. This over-parametrization makes it possible to model scenes containing rigid and nonrigid motions, and as is presented in Section 4.4, it includes and generalizes other motion representations.

4.1 Twist-motion representation

Every 3D point $\mathbf{X}$, or scene point, is assumed to belong to a rigid surface $\mathbf{S}$, which is free to move between two time steps. Let ξ be the 6-vector twist representing the rigid motion of the surface $\mathbf{S}$. Under the twist action, every surface point performs a 3D motion and the assemblage of motions of all surfaces generates the observed scene flow. Instead of directly representing the elements of the motion field as 3D vectors, we use an over-parametrized model to describe the motion of every point as the rigid transformation of the surface. The 3D motion field of the scene is embedded in a *twist motion field*, $\xi(\mathbf{x}) : \mathbf{\Omega} \to \mathbb{R}^6$, where the motion of a 3D point $\mathbf{X}$, between two time steps, is modeled by:

$$\tilde{\mathbf{X}}_{t+1} = e^{\hat{\xi}(\mathbf{x}_t)} \tilde{\mathbf{X}}_t \,. \tag{4.1}$$

In the twist-motion representation, every scene point follows a rigid motion, which is represented by 6 parameters. This way is possible to encourage piecewise smooth or constant solutions on this field of twist motions. This representation exploits the local rigidity of the scene in order to better approximate complex 3D motions. Nevertheless, this representation does not expect any a priori semi-rigid configuration enabling a flexible formulation of the scene flow estimation, as is presented in Chapter 5.

Between any two time steps, the scene flow, $\mathbf{v}(\mathbf{x})$, can be directly recovered from a given twist motion field, $\xi(\mathbf{x})$, as follows:

$$\begin{pmatrix} \mathbf{v}(\mathbf{x}) \\ 1 \end{pmatrix} = \left(e^{\hat{\xi}(\mathbf{x})} - \mathbf{I}_{4\times 4} \right) \tilde{\mathbf{X}} \,. \tag{4.2}$$

4.1.1 Twist motion on the image

Under the action of a twist ξ between t and $t+1$, the motion of every 3D point induces an *image flow* (u, v) in its RGBD representation. Changes and invariances of the RGBD data can be used to solve for the scene flow, as is described in Section 5.2. For this purpose, it is useful to define a warping function that allows the constraint of a rigid motion in the image domain, enabling the exploitation of color and depth images. Particularly, because the twist representation, this warp is required to map the twist action to the image domain. Let $\mathbf{W}_\xi(\mathbf{x}, \xi) : \mathbb{R}^2 \times \mathbb{R}^6 \to \mathbb{R}^2$ be the warping function that maps each (non-occluded) scene point to its new location after the rigid motion ξ. The warping function is defined as:

$$\mathbf{W}_\xi(\mathbf{x}, \xi) = \pi \left(e^{\hat{\xi}} \tilde{\mathbf{X}} \right), \quad \text{with} \quad \tilde{\mathbf{X}} = \begin{pmatrix} \pi^{-1}(\mathbf{x}, Z(\mathbf{x})) \\ 1 \end{pmatrix}. \tag{4.3}$$

Accordingly, under the action of twist motion ξ, a 3D point $\mathbf{X}_t$ becomes $\mathbf{X}_{t+1}$, and its projection on the image is modeled by:

$$\mathbf{x}_{t+1} = \mathbf{W}_\xi(\mathbf{x}_t, \xi). \tag{4.4}$$

Let $S^k(\mathbf{x})$ be a component of the RGBD image, e.g., brightness or depth, evaluated at $\mathbf{x} \in \mathbf{\Omega}$, and let $\rho_{S^k}(\mathbf{x}, \xi)$ be a differentiable consistency function between $S^k_t(\mathbf{x})$ and $S^k_{t+1}(\mathbf{W}_\xi(\mathbf{x}, \xi))$. Every twist motion vector ξ is solved aiming to minimize one or multiple consistency functions like $\rho_{S^k}(\mathbf{x}, \xi)$. Optimization problems on manifolds, such as $SE(3)$, can be solved by calculating incremental steps in the tangent space to the manifold. Particularly, the tangent space of the Lie group $SE(3)$ corresponds to its Lie algebra $se(3)$. If a gradient-based procedure is used, it requires computation of the derivatives of the warp with respect to the twist motion ξ, where $\hat{\xi} \in se(3)$, see Equation (2.8). The warp (4.3) is nonlinear in ξ yielding to a complicated expression for its Jacobian. However, this can be well approximated by a linear approximation around any small neighborhood on the manifold, as is presented below. The definition of this linearized approximation simplifies the optimization procedure providing a simple expression of the Jacobian, which benefits the efficiency of any solver. Moreover, the linearization makes it possible to analyze how each component of the twist motion affects the motion in the image domain.

4.1.2 Linearization of the warping function

Let $\mathbf{x}$ be the projection on the image of a given 3D point $\mathbf{X}$. Under the action of twist ξ, the pixel $\mathbf{x}$ becomes $\mathbf{x}_\xi = \mathbf{W}_\xi(\mathbf{x}, \xi)$. Moving away from ξ, on the manifold,

induces an additional image flow on the projection of the 3D point. Let $\Delta\xi$ be a small twist increment, i.e, close to the identity in $SE(3)$, applied from ξ, yielding to the equivalent twist motion $\xi_\Delta = \log(e^{\hat{\Delta\xi}}e^{\hat{\xi}})$. The projection of 3D point $\mathbf{X}$, under the action of ξ_Δ, can be expressed using the warping function as follows:

$$\mathbf{W}_\xi(\mathbf{x}, \log(e^{\hat{\Delta\xi}}e^{\hat{\xi}})) = \mathbf{W}_\xi(\mathbf{x}_\xi, \Delta\xi) = \mathbf{x}_\xi + \delta\mathbf{x}(\mathbf{x}_\xi, \Delta\xi), \tag{4.5}$$

where $\delta\mathbf{x}(\mathbf{x}_\xi, \Delta\xi)$ is the image flow induced by the twist increment $\Delta\xi$. Knowing that $\Delta\xi$ is close to the identity, the twist induces a small rotation and so that the exponential function can be well approximated by the linear expression $e^{\hat{\Delta\xi}} \approx \mathbf{I} + \hat{\Delta\xi}$, see Section 2.2.2. Therefore, the image flow can be written as follows:

$$\begin{aligned}\delta\mathbf{x}(\mathbf{x}_\xi, \Delta\xi) &= \pi\left(e^{\hat{\Delta\xi}}\tilde{\mathbf{X}}_\xi\right) - \pi\left(\tilde{\mathbf{X}}_\xi\right) \\ &\approx \pi\left((\mathbf{I} + \hat{\Delta\xi})\tilde{\mathbf{X}}_\xi\right) - \pi\left(\tilde{\mathbf{X}}_\xi\right),\end{aligned} \tag{4.6}$$

with $\mathbf{X}_\xi = \pi^{-1}(\mathbf{x}_\xi, Z(\mathbf{x}_\xi))$, and where we abuse of notation of π to project 3D points directly in homogeneous coordinates. Applying the twist and the projective function in (4.6), we have:

$$\delta\mathbf{x}(\mathbf{x}_\xi, \Delta\xi) \approx \begin{pmatrix} f_x\frac{X_\xi+\Delta X}{Z_\xi+\Delta Z} + c_x \\ f_y\frac{Y_\xi+\Delta Y}{Z_\xi+\Delta Z} + c_y \end{pmatrix} - \begin{pmatrix} f_x\frac{X_\xi}{Z_\xi} + c_x \\ f_y\frac{Y_\xi}{Z_\xi} + c_y \end{pmatrix}, \tag{4.7}$$

with $(\Delta X, \Delta Y, \Delta Z)^T$ the vector that contains the components of the 3D displacement induced by the twist $\Delta\xi$, which are given by:

$$\begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix} = \begin{pmatrix} 0 & -Z_\xi & Y_\xi \\ Z_\xi & 0 & -X_\xi \\ -Y_\xi & X_\xi & 0 \end{pmatrix} \begin{pmatrix} \Delta\omega_X \\ \Delta\omega_Y \\ \Delta\omega_Z \end{pmatrix} + \begin{pmatrix} \Delta\tau_X \\ \Delta\tau_Y \\ \Delta\tau_Z \end{pmatrix}. \tag{4.8}$$

Equation (4.9) can be simplified if the Z-component of the 3D motion is very small with respect to the depth of the 3D point. By assuming that $|\Delta Z/Z| \ll 1$ then $Z_\xi(Z_\xi + \Delta Z) \approx {Z_\xi}^2$, and the induced flow becomes:

$$\delta\mathbf{x}(\mathbf{x}_\xi, \Delta\xi) \approx \begin{pmatrix} f_x\frac{X_\xi\Delta Z + Z_\xi\Delta X}{Z_\xi^2} \\ f_y\frac{Y_\xi\Delta Z + Z_\xi\Delta Y}{Z_\xi^2} \end{pmatrix}. \tag{4.9}$$

Therefore, by defining $\mathbf{J}(\mathbf{x}_\xi)$ as the Jacobian matrix evaluated at $\mathbf{x}_\xi$, as follows:

$$\mathbf{J}(\mathbf{x}_\xi) = \begin{pmatrix} \frac{fx}{Z_\xi} & 0 & -\frac{x_\xi}{Z_\xi} & -\frac{x_\xi y_\xi}{f_y} & \frac{f_x+x_\xi^2}{f_y} & -\frac{y_\xi f_x}{f_y} \\ 0 & \frac{fy}{Z_\xi} & -\frac{y_\xi}{Z_\xi} & -\frac{f_y+y_\xi^2}{f_x} & \frac{x_\xi y_\xi}{f_x} & \frac{x_\xi f_y}{f_x} \end{pmatrix}, \tag{4.10}$$

and given a small twist increment $\Delta\xi$, the warping function can be well approximated around the twist ξ, by the following linear version:

$$\mathbf{W}_\xi(\mathbf{x}, \log(e^{\hat{\Delta\xi}} e^{\hat{\xi}})) = \mathbf{W}_\xi(\mathbf{x}, \xi) + \mathbf{J}(\mathbf{x}_\xi)\Delta\xi. \tag{4.11}$$

4.2 3D-motion representation

The scene flow can be represented directly as a 3D motion field, which is its original definition in Section 2.1.2. In this representation, every 3D point $\mathbf{X}$ is also assumed to belong to a moving rigid surface $\mathbf{S}$, but instead of an explicit representation of the rigid motion, the 3D displacement of every surface points is considered. Let $\mathbf{x}$ be the projection of $\mathbf{X}$. Under the motion of the surface, the 3D displacement of $\mathbf{X}$ can be written as $\mathbf{v}(\mathbf{x}) = (v_X, v_Y, v_Z)$. The assemblage of all motions forms the scene flow, $\mathbf{v}(\mathbf{x}) : \mathbf{\Omega} \to \mathbb{R}^3$, where the motion of every 3D point, between two time steps, is modeled by:

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \mathbf{v}(\mathbf{x}_t)\,. \tag{4.12}$$

In the 3D-motion representation, every scene point performs a 3D displacement that is directly represented by 3 parameters. Under this representation, the local rigidity assumption cannot be directly exploited. However it is still possible to encourage smooth piecewise solutions, favoring semi-rigid motions. Moreover, it is also possible to easily fit a local rigid motion, which can be useful to encourage semi-rigid solutions, as is presented in Section 5.3.2.

4.2.1 3D motion on the image

The 3D motion of a scene point induces an *image flow*, (u, v), in its projection on the image. The definition of a warping function makes it possible to constrain every 3D motion vector on the RGBD images. In order to define the 3D-motion warp, we use the twist warping function (4.3), by considering the 3D displacement as the particular case of the twist motion, where the rotational component is zero. Let $\mathbf{W}_\mathbf{v}(\mathbf{x}, \mathbf{v}) : \mathbb{R}^2 \times \mathbb{R}^3 \to \mathbb{R}^2$ be the warping function that maps every scene point into its new location, after the action of 3D motion $\mathbf{v}$, as follows:

$$\mathbf{W}_\mathbf{v}(\mathbf{x}, \mathbf{v}) = \pi\left(\mathbf{X} + \mathbf{v}\right), \quad \text{with} \quad \mathbf{X} = \pi^{-1}(\mathbf{x}, Z(\mathbf{x})). \tag{4.13}$$

Therefore, if between two time steps, the point $\mathbf{X}_t$ performs a 3D motion $\mathbf{v}$, becoming $\mathbf{X}_{t+1}$, its projection on the image is modeled by:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{W}_\mathbf{v}(\mathbf{x}_t, \mathbf{v}). \tag{4.14}$$

4.2.2 Linearization

The warping function (4.13) can be linearized around a given 3D motion $\mathbf{v}$. Under 3D motion $\mathbf{v}$, scene point $\mathbf{x}$ becomes $\mathbf{x}_\mathbf{v} = \mathbf{W}_\mathbf{v}(\mathbf{x}, \mathbf{v})$. Let $\Delta\mathbf{v} = (\Delta v_X, \Delta v_Y, \Delta v_Z)^T$

be a small 3D motion increment, then the projection of the 3D point $\mathbf{X}$, after the motion given by $\mathbf{v} + \Delta\mathbf{v}$, can be expressed using the warping function as follows:

$$\mathbf{W}_{\mathbf{v}}(\mathbf{x}, \mathbf{v} + \Delta\mathbf{v}) = \mathbf{x}_{\mathbf{v}} + \delta\mathbf{x}(\mathbf{x}_{\mathbf{v}}, \Delta\mathbf{v}), \tag{4.15}$$

with $\delta\mathbf{x}(\mathbf{x}_{\mathbf{v}}, \Delta\mathbf{v})$ the image flow induced by $\Delta\mathbf{v}$. Knowing that $\mathbf{v}$ can be seen as a twist with a zero rotational component, we use the result (4.10), from Section 4.1.2. By defining $\mathbf{J}(\mathbf{x}_{\mathbf{v}})$ as the Jacobian matrix evaluated at $\mathbf{x}_{\mathbf{v}}$ as follows:

$$\mathbf{J}(\mathbf{x}_{\mathbf{v}}) = \frac{1}{Z_{\mathbf{v}}} \begin{pmatrix} fx & 0 & -x_{\mathbf{v}} \\ 0 & fy & -y_{\mathbf{v}} \end{pmatrix}, \tag{4.16}$$

and given a small increment $\Delta\mathbf{v}$, the warping function can be well approximated around $\mathbf{v}$, by the following linear version:

$$\mathbf{W}_{\mathbf{v}}(\mathbf{x}, \mathbf{v} + \Delta\mathbf{v}) = \mathbf{W}_{\mathbf{v}}(\mathbf{x}, \mathbf{v}) + \mathbf{J}(\mathbf{x}_{\mathbf{v}})\Delta\mathbf{v}. \tag{4.17}$$

4.3 Other motion representations

4.3.1 Rigid body model

The inter-frame image flow induced by the 3D motion of a rigid surface has been modeled in the literature using different assumptions. For brevity, let the focal lengths, $\{f_x, f_y\}$, be the unit and the optical center, (c_x, c_y), be at the image origin. The instantaneous velocity of a retinal point (x, y), corresponding to a given 3D point with known depth Z, is modeled by Longuet-Higgins and Prazdny (1980) as:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} 1 & 0 & -x \\ 0 & 1 & -y \end{pmatrix} \begin{pmatrix} t_X \\ t_Y \\ t_Z \end{pmatrix} - \begin{pmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{pmatrix} \begin{pmatrix} \omega_X \\ \omega_Y \\ \omega_Z \end{pmatrix}, \tag{4.18}$$

with $(u, v) = (\partial x/\partial t, \partial y/\partial t)$, and where $\mathbf{t} = (t_X, t_Y, t_Z)^T$ and $\omega = (\omega_X, \omega_Y, \omega_Z)^T$ are the instantaneous translation and angular velocity vectors, respectively. Horn and Weldon (1988) use this instantaneous velocity model to describe the inter-frame pixel motion, where velocities become displacements between t and $t+1$, and the optical flow is defined as $(u, v) = (x_{t+1} - x_t, y_{t+1} - y_t)$. Adiv (1985) shows that equation (4.18) approximates the image flow induced by a rigid motion under the assumptions that the view of field of the camera is not very large, and V_Z/Z ratio and rotational velocities are very small. This model is named *rigid body model* in the hierarchy classification of motions proposed by Bergen *et al.* (1992), and it is characterized by the 6 degrees of freedom (DOF) of the rigid motion. Also, this model coincides with the linearized warping function of the twist motion (4.11) and is only valid to capture the image motion for rigid motions close to the identity.

4.3.2 Affine flow model

When the distance between the background surfaces and the camera is large, it is possible to approximate the image motion as an affine transformation (Bergen *et al.*, 1992) using the following model:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{pmatrix}. \tag{4.19}$$

This *affine flow model* can be seen as the particular case of the *rigid body model* with only 4 DOF, where the rotational components ω_x and ω_y are assumed to be negligible. Following equation (4.18) we can write the image flow as:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} 1 & 0 & -x \\ 0 & 1 & -y \end{pmatrix} \begin{pmatrix} t_X \\ t_Y \\ t_Z \end{pmatrix} + \begin{pmatrix} -y \\ x \end{pmatrix} \omega_Z. \tag{4.20}$$

This way the induced flow is modeled by the coefficients $\{a_1, ..., a_6\}$, which are functions of the 3D displacement (t_X, t_Y, t_Z) and the rotational component ω_Z.

4.3.3 Planar surface flow model

Equation (4.18) assumes that the depth of every pixel is known, which is not always the case for many applications. If the optical flow corresponds to the projection of a planar surface, $k_X X + k_Y Y + k_Z Z = 1$, the plane parameters (k_X, k_Y, k_Z) constrain the flow induced by the 3D motion of the surface. The depth variable in (4.18) can be removed by using the plane equation, thus the image flow is expressed as a function of 8 parameters (Adiv, 1985):

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} a_1 + a_2 x + a_3 y + a_7 x^2 + a_8 xy \\ a_4 + a_5 x + a_6 y + a_7 xy + a_8 y^2 \end{pmatrix}, \tag{4.21}$$

where coefficients $\{a_1, ..., a_8\}$ depend on the rigid motion parameters $(t_X, t_Y, t_Z)^T$ and $(\omega_X, \omega_Y, \omega_Z)^T$ and the surface parameters $(k_X, k_Y, k_Z)^T$. This 8 DOF optical flow model is called *planar surface flow model* by Bergen *et al.* (1992). Also, this model is known as the *homography model* by Baker and Matthews (2004).

4.3.4 Orthographic camera models

Rigid body and affine models depend on depth data to compute the induced motion. Therefore they make it possible to use depth data jointly with RGB information to constrain motion in the image domain. However, it is also possible to consider a

simpler camera model, as an orthographic camera, which is the most widely used assumption in optical flow methods. In this case, the scene flow is inferred from the computed optical flow and the provided depth data. Considering an orthographic camera model allows to assess the contribution of incorporating depth data in the 2D motion constraint.

The action of a rigid transformation $\mathbf{G} = \{\mathbf{R}, \mathbf{t}\}$ can be written as $\mathbf{X}_{t+1} = \mathbf{R}\mathbf{X}_t + \mathbf{t}$, so that the orthographic projection of the resulting 3D point $\mathbf{X}_{t+1}$ is given by:

$$\pi_{\mathbf{o}}(\mathbf{X}_{t+1}) = \begin{pmatrix} R_{11}X_t + R_{12}Y_t + R_{13}Z_t \\ R_{21}X_t + R_{22}Y_t + R_{23}Z_t \end{pmatrix} + \begin{pmatrix} t_X \\ t_Y \end{pmatrix}, \tag{4.22}$$

where the orthographic projection $\pi_{\mathbf{o}} : \mathbb{R}^3 \to \mathbb{R}^2$ maps a 3D point $\mathbf{X} = (X, Y, Z)^T$ to a pixel $\mathbf{x} = (x, y)^T$ on $\mathbf{\Omega}$, by:

$$\pi_{\mathbf{o}}(\mathbf{X}) = (k_x X + c_x, k_y Y + c_y)^T, \tag{4.23}$$

with k_x and k_y parameters of the camera, and (c_x, c_y) its principal point. For the sake of simplicity, we assume that $k_x = k_y = 1$ and $c_x = c_y = 0$. Therefore, the induced image flow, $\mathbf{u} = (u, v)^T$, between t and $t + 1$, is given by:

$$\mathbf{u} = \pi_{\mathbf{o}}(\mathbf{X}_{t+1}) - \pi_{\mathbf{o}}(\mathbf{X}_t), \tag{4.24}$$

and can be represented by the following model:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} (R_{11} - 1)\, x + R_{12}\, y + (R_{13}Z + t_X) \\ R_{21}\, x + (R_{22} - 1)\, y + (R_{23}Z + t_Y) \end{pmatrix}, \tag{4.25}$$

which corresponds to the affine motion model. In this case, the depth information Z is irrelevant and is combined with the independent coefficients a_1 and a_4 of the affine model, see equations (4.19) and (4.20). This 4 DOF model is used by Nir *et al.* (2008) to compute an over-parametrized optical flow. A simpler model is defined by assuming a negligible rotation with an orthographic camera model. In this simplified model the image flow is given by:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} t_X \\ t_Y \end{pmatrix}, \tag{4.26}$$

where the 3D surfaces are assumed to only perform 3D translations and be captured by an orthographic camera. This 2 DOF model, which we name 2D-motion representation, is the most used in the optical flow literature. The warping function, $\mathbf{W}_{\mathbf{u}}(\mathbf{x}, \mathbf{u}) : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$, is defined to map a pixel to its new location as a function

of the perceived optical flow $\mathbf{u}$, according to:

$$\mathbf{W_o}(\mathbf{x}, \mathbf{u}) = \mathbf{x} + \mathbf{u}. \tag{4.27}$$

The scene flow, $\mathbf{v}(\mathbf{x})$, can be inferred using the optical flow $\mathbf{u}$ and the depth image $Z(\mathbf{x})$, as follows:

$$\mathbf{v}(\mathbf{x}) = \pi^{-1}(\mathbf{x} + \mathbf{u}, Z(\mathbf{x} + \mathbf{u})) - \pi^{-1}(\mathbf{x}, Z(\mathbf{x})). \tag{4.28}$$

4.4 Summary of motion representations

In this thesis, we consider three different motion representations, as is shown in Table 4.1. We are mainly concerned in the proposed twist-motion representation, which enables the direct use of the semi-rigid properties of the scene for the 3D motion field estimation. We also use the 3D-motion representation, which allows the direct solution of the scene flow and can be seen as particular case of the twist-motion representation. Both twist-motion and 3D-motion representations exploit depth data through a projective warping function. On the other hand, the 2D-motion representation considers an orthographic camera, neglecting the depth changes in the constraint of motion in the image, and yielding to a simpler formulation of the motion estimation problem. Although every representation have its own warping function, for the sake of simplicity, the generic expression $\mathbf{W}(\mathbf{x}, \cdot)$ is used, and the parameter ξ, $\mathbf{v}$, or $\mathbf{u}$ is used to identify the respective motion representation.

Representation	**Warping function**	**Small-increment model**	**Scene flow**
6-Twist	$\mathbf{W}_\xi(\mathbf{x}, \xi) = \pi\left(e^{\hat{\xi}}\tilde{\mathbf{X}}\right)$	$\mathbf{W}_\xi(\mathbf{x}, \Delta\xi) = \mathbf{x} + \mathbf{J}_\xi(\mathbf{x})\Delta\xi$	$\left(e^{\hat{\xi}} - \mathbf{I}_{4\times4}\right)\tilde{\mathbf{X}}$
3D-motion	$\mathbf{W}_\mathbf{v}(\mathbf{x}, \mathbf{v}) = \pi\left(\mathbf{X} + \mathbf{v}\right)$	$\mathbf{W}_\mathbf{v}(\mathbf{x}, \Delta\mathbf{v}) = \mathbf{x} + \mathbf{J}_\mathbf{v}(\mathbf{x})\Delta\mathbf{v}$	$\mathbf{v}$
2D-motion	$\mathbf{W_o}(\mathbf{x}, \mathbf{u}) = \mathbf{x} + \mathbf{u}$	$\mathbf{W_o}(\mathbf{x}, \Delta\mathbf{u}) = \mathbf{x} + \Delta\mathbf{u}$	$\pi^{-1}(\mathbf{x} + \mathbf{u}) - \pi^{-1}(\mathbf{x})$

Table 4.1: *Summary of motion representations.*

5 Scene Flow Model

The RGBD representation of a scene changes when the observed 3D surfaces move or the camera undergoes motion, or both simultaneously. These changes in RGBD data, together with the definition of a set of invariants of the representation, allow the estimation of the 3D motion field presented in the scene. For example, the color patterns of a given surface are expected to remain unchanged when it moves in the scene, but also its color must differ from that of other surfaces, in order to perceive its motion in the RGB data. Similarly, when a rigid surface moves in the scene, it is expected that its 3D structure does not change and its intrinsic properties remain consistent in the observed depth data. However, at the same time, the representation should be able to capture the relative motion between the surface and the sensor. Figures 5.1 and 5.2 present examples of changes and invariances in the RGBD representation when surfaces move relative to the sensor. The definition of an appropriate RGBD-based framework for scene flow estimation makes it possible to take advantage of both sources of data: color and depth. However, there are several ways to define the scene flow model and different assumption can be used on the scene, in order to exploit the RGBD data. We base our framework on the assumption of a semi-rigid scene and on the use of constancy constraints directly on the RGBD images. Below we describe below our approach to exploit color and depth data for scene flow estimation, using the twist- and 3D-motion representations.

The scene flow estimation problem is formulated as the determination of the inter-frame 3D motion between two time steps. Accordingly, given two pairs of RGBD images $\{I_{c1}, Z_1\}$ and $\{I_{c2}, Z_2\}$ the goal is to solve for the 3D motion field that best explains the observed data, aiming to determine the real 3D motion executed in the scene. The use of an appropriate representation of motion makes possible to project 3D motions into the image domain and define consistency assumptions on color and depth data. This way plausible 3D-motion vectors can be solved to satisfy or minimize one or several RGBD-based constraints.

Fig. 5.1: *RGBD changes using a fixed camera. Form left to right, top to bottom: (a) first image, (b) second image, and absolute difference in (c) intensity and (d) depth. Moving surfaces produce changes in intensity and depth, while static scene points remain unchanged, except by the noise.*

A motion constraint example

Let $I^k(\mathbf{x})$ be a color component of a scene point $\mathbf{X}$, after an inter-frame 3D motion $\mathbf{v}$, it is intuitive to assume that component I^k remains unchanged, yielding to the constraint:

$$I_1^k(\mathbf{x}) = I_2^k(\pi(\mathbf{X} + \mathbf{v})). \tag{5.1}$$

This constancy constraint is not linear on $\mathbf{v}$, but can be linearized, using a first-order Taylor series expansion, to incrementally solve for the 3D motion. This linearization assumes that the image changes linearly with the projection of $\mathbf{X} + \mathbf{v}$, which is not always the case, especially under large motions in the image domain. This effect can be reduce by solving iteratively for small increments of $\mathbf{v}$, where the linearization is more reliable. However, it is clear that only one equation as (5.1) does not uniquely determine the 3D motion $\mathbf{v}$ for a given scene point $\mathbf{X}$: one equation with three unknowns. Since the full RGB information is available it seems reasonable to define similar constancy constraints to exploit the three color components. Nevertheless, the gradient information of the RGB component is highly correlated yielding in most cases to a ill conditioned system of equations that do not allow a reliable estimation of motion. Such limitation is known as the *aperture problem* in optical flow estimation and implies that only the component of the 2D motion, normal to

Fig. 5.2: *RGBD changes with a moving camera. From left to right, top to bottom: (a) first image, (b) second image, and absolute difference in (c) intensity and (d) depth. The movement of the camera, as well as that of the person, produce a 3D motion of every scene point. Particularly, the head turning produces more changes in intensity than in depth, while the camera motion is better perceived on depth data in some regions, such as neck and t-shirt.*

the gradient direction, can be determined from a single constraint. This disability is not exactly the same in the scene flow estimation problem for two main reasons: *i*) a 3D representation of the scene is provided by the depth data and *ii*) we aim to solve for a 3D motion rather than simply the 2D motion on the image. Therefore, it is evident that at least a second assumption is needed. The depth data can be used to formulate an additional constraint to ensure the consistency of the 3D motion with the 3D structure of the scene, as is presented in Section 5.2.2. Solving for the 3D motion using single constraints on depth and a color images, yields a kind of *3D aperture problem*. Motion in the image domain can be determined only if is normal to the gradient direction in color or depth images. Moreover, color and depth gradient information could coincide yielding to the 2D aperture problem. Conversely, any change in the Z-component of the motion can be determined since it is directly reflected on the depth data, only limited by physics restrictions of the depth sensor. Besides, vanishing gradients, noise and outliers are also very common in color and depth images, making of the 3D motion computation an ill-posed problem that cannot be solved independently for each point.

We present below a variational framework to estimate scene flow by exploiting local and global properties of semi-rigid scenes.

5.1 Scene Flow Energy

In order to get a reliable estimation of the scene flow, it is required to impose a set of assumptions on the 3D motion field. In the case of optical flow estimation, mostly a *local* or *global* regularity is used to model motion. On one hand, a *local method* assumes that the motion in a local neighborhood on the image is constant. This way it is possible to locally solve for the motion that best explains the local data, ignoring the surrounding information. This idea dates from the seminal work by Lucas and Kanade (1981) for stereo registration, which have been widely used in optical flow. On the other hand, a *global method* assumes that the motion field is smooth. In their seminal work Horn and Schunck (1981) introduce a global variational approach, where a (global) dependency is defined on the motion at every pixel with those at all the other pixels. Although local methods are robust to noise and can handle larger displacements, current global variational methods are by far the top performer in motion estimation tasks (Sun *et al.*, 2010). Nevertheless, as is proposed by Bruhn *et al.* (2005), it is also possible to benefit from local and global approaches in the same formulation. We follow this belief to exploit the semi-rigidity properties of real scenes from RGBD images. For this purpose, we use local methods to take advantage of the local rigidity of the scene while global methods are used to encourage piecewise smooth solutions, which are expected to be observed due to the assumed piecewise rigidity of the scene. Accordingly, we formulate the scene flow computation as the minimization problem of the energy:

$$E(\xi) = E_D(\xi) + \alpha E_S(\xi), \tag{5.2}$$

where the twist-motion representation is used, aiming to better exploit the semi-rigidity of the scene and which, at the same time, being able to cover other motions models. The first term, $E_D(\xi)$, is the *data term*, which measures how consistent is the estimated scene flow with the observed intensity and depth data. The *smoothness term*, $E_S(\xi)$, is defined to favor smooth motion fields but preserving motion discontinuities. The local rigidity assumption can be exploited both in the data and smoothness terms. By using this formulation is possible to get an adjustable combination between local and piecewise rigidity in the solution. We describe below the definition of each of components of the scene flow energy, aiming to exploit the RGBD representation of the scene.

5.2 Data term

Our goal is to solve for the scene flow that best explains the observed color and depth data. The definition of the data term is based in two main assumptions: *i*) the color of the scene points remains (approximately) unchanged and *ii*) the depth data is able to reliably capture the 3D structure of the scene at every time step.

In order to constrain motion in the RGBD data the scene flow has to be projected to the image domain. Considering the twist-motion representation, the warping function given by Equation (4.3), models the motion of each scene point, enabling the formulation of RGBD constraints for elements of this twist field on the image.

5.2.1 Color-based constraint

As in most optical flow methods, we exploit the color invariance through the gray value image by defining a *brightness constancy assumption* (*BCA*):

$$I_2(\mathbf{W}(\mathbf{x}, \xi)) = I_1(\mathbf{x}), \tag{5.3}$$

where $I(\mathbf{x})$ represent the gray value of pixel $\mathbf{x}$. Although it is also possible to define constraints by using each of the RGB components or another color space, the BCA has proved to be simple and effective. The main drawback of the BCA is its sensitivity to illumination changes, which are common in real scenes. Moreover, current RGBD sensors perform automatic white balance and exposure control by default, which can alter gray values when the composition of the scene change suddenly or when surfaces approach or move away from the camera. Therefore the invariance expected in (5.3) is often violated. In order to gain robustness against slight changes in brightness, Brox *et al.* (2004) define a gradient invariance of the gray value image. However this assumption fails under rotations, especially around the Z-axis since the gradient orientation changes. Nevertheless, the magnitude of the gradient, $I^g(\mathbf{x}) = |\nabla I(\mathbf{x})|$, is expected to remain unchanged under a rotation, and in general under any rigid motion. Accordingly, we define a *gradient constancy assumption* (*GCA*) as:

$$I_2^g(\mathbf{W}(\mathbf{x}, \xi)) = I_1^g(\mathbf{x}). \tag{5.4}$$

Figure 5.3 shows an example of how changes in the gray value image, not related with the motion in the scene, affects brightness and gradient images.

5.2.2 Depth-based constraint

Variations of the 3D structure, induced by the scene flow, must be consistent with the observed depth data at any time step. Therefore, it is possible to define a *depth variation constraint* (*DVC*) as follows:

$$Z_2(\mathbf{W}(\mathbf{x}, \xi)) = Z_1(\mathbf{x}) + \delta_Z(\mathbf{x}, \xi), \tag{5.5}$$

where $\delta_Z(\mathbf{x}, \xi)$ denotes the variation in depth induced on the 3D point $\pi^{-1}(\mathbf{x}, Z_1(\mathbf{x}))$ by the twist motion ξ. Let $\delta_{\mathbf{3D}}(\mathbf{x}, \xi) = (\delta_X, \delta_Y, \delta_Z)^T$ represent the 3D variation

Fig. 5.3: *Brightness constancy violation. From left to right, top to bottom: (a) first image, (b) second image, (c) image difference, (d) first and (e) second images of the magnitude of the gradient, and (f) their difference. While the camera is moving, there exists an automatic white balance that modifies the gray value image, as can be noted in the image difference (c), changes that are not related to the motion. On the other hand, the magnitude of the gradient is too little affected since it measures the image structure, or texture, regardless uniform changes in luminance.*

induced on $\mathbf{x}$ by twist motion ξ, defined as:

$$\delta_{\mathbf{3D}}(\mathbf{x}, \xi) = \left(e^{\hat{\xi}} - \mathbf{I}_{4\times 4}\right) \tilde{\mathbf{X}}, \qquad \text{with} \quad \tilde{\mathbf{X}} = \begin{pmatrix} \pi^{-1}(\mathbf{x}, Z_1(\mathbf{x})) \\ 1 \end{pmatrix}. \tag{5.6}$$

Accordingly, the variation in depth, δ_Z, is obtained from the third component of vector $\delta_{\mathbf{3D}}$. This constraint enforces the consistency between the motion captured by the depth sensor and the estimated motion.

3D-motion case. If the 3D-motion representation is used for the scene flow, $\mathbf{v} = (v_X, v_Y, v_Z)^T$, the DVC takes a simpler form, which is given by:

$$Z_2(\mathbf{W}(\mathbf{x}, \mathbf{v})) = Z_1(\mathbf{x}) + v_Z, \tag{5.7}$$

reflecting that the third component of the 3D motion field must be consistent with the observed change in depth.

5.2.3 Consistency measure

Different factors make the measurement of these consistency constraints a challenge. First, the presence of noise on color and depth data produces perturbations in the

RGBD data, frequently violating the consistency terms. Second, lens distortions and the inaccuracy of the depth measures may bring errors in the projection of the 3D motion on the image. Third, partial or total occlusions of the 3D surfaces have to be handle since consistency constraints are not valid at these regions and could strongly affect the motion estimation. Finally, external changes uncorrelated to the 3D motion, as illumination changes, specular surfaces and shadows, produce inconsistency between the RGBD representation of the scene and the scene flow. For these reasons, RGBD based constraints are not obeyed in general and it is required to measure how consistent is a given estimation with the observed data. Let ρ_I, ρ_g and ρ_Z be the residuals of BCA, GCA and DVC constraints, respectively, given by:

$$\rho_I(\mathbf{x},\xi) = I_2(\mathbf{W}(\mathbf{x},\xi)) - I_1(\mathbf{x}), \tag{5.8}$$

$$\rho_g(\mathbf{x},\xi) = I_2^g(\mathbf{W}(\mathbf{x},\xi)) - I_1^g(\mathbf{x}), \tag{5.9}$$

$$\rho_Z(\mathbf{x},\xi) = Z_2(\mathbf{W}(\mathbf{x},\xi)) - (Z_1(\mathbf{x}) + \delta_Z(\mathbf{x},\xi)). \tag{5.10}$$

In order to cope with outliers in brightness and depth brought by noise, occlusions or motion inconsistencies, a robust norm is required. In contrast to quadratic penalties that give too much influence to outliers in the motion estimation, l_1 norm has proved to be very effective to deal with outliers in the data term in global optical flow methods (Wedel *et al.*, 2008). We use the Charbonnier penalty $\Psi\left(s^2\right) = \sqrt{s^2 + 0.001^2}$, which is a differentiable approximation of the absolute value, see Figure 2.9. This penalty is applied jointly to BCA and GCA, and separately to DVC, to get the total RGBD consistency measure:

$$\rho_{data}(\mathbf{x},\xi) = \Psi\left(\rho_I^2(\mathbf{x},\xi) + \gamma\rho_g^2(\mathbf{x},\xi)\right) + \lambda\Psi\left(\rho_Z^2(\mathbf{x},\xi)\right), \tag{5.11}$$

where the constant γ balances brightness and gradient residuals, while λ weights intensity and depth terms.

5.2.4 Local consistency encouragement

It is well known that local methods are often more robust under noise than global methods, allowing reliable motion estimations even facing contradictory or missing data. Although local methods allow fast sparse solutions, they fail to provide dense estimations since the surrounding information is not exploited to disambiguate for motion in untextured regions. Local methods properties can benefit and complement the global motion estimation, as is proposed by Bruhn *et al.* (2005).

Most scenes of interest can be well modeled as locally rigid scenes, i.e. they can be seen as scenes composed by independent 3D rigid parts. Under this context, the local assumption can benefit the motion estimation. However, the local assumption can bias the estimation if the representation of motion does not properly model these

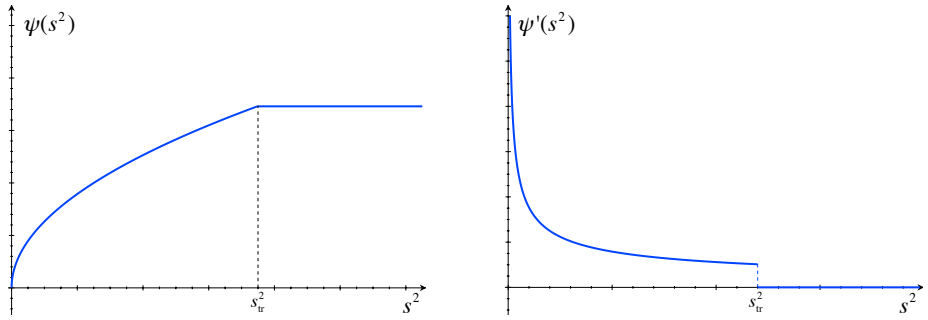


Fig. 5.4: *Truncated Charbonnier penalty. From left to right: (a) truncated version of the robust norm* $\Psi\left(s^2\right) = \sqrt{s^2 + 0.001^2}$*, for a given* S^2_{tr}*, and (b) its derivative. This truncated penalty drops all square residues larger than* S^2_{tr} *from the local estimation.*

local properties. For example, under the rotation of a small rigid surface, neither the optical flow nor the 3D motion field should be straightforward encouraged to be locally consistent since motion patterns differ between them. Conversely, the twist representation is able to reflect the local rigidity of the scene on the image and a local estimation allows to directly exploit this properties. Therefore we define a local consistency error ρ^l_{data}, which measures how well a given twist ξ locally satisfies the RGBD constraints, according to:

$$\rho^l_{data}\left(\mathbf{x}, \xi\right) = \sum_{\mathbf{x}' \in N(\mathbf{x})} w(\mathbf{x}')\, \rho_{data}\left(\mathbf{x}', \xi\right) \tag{5.12}$$

where $N(\mathbf{x})$ is an image neighborhood centering on $\mathbf{x}$, and w is a specific weighting function. The local neighborhood $N(\mathbf{x})$ can be setting to a fix size or adjusted as a function of the depth in order to cover a constant size in the scene. It is clear that if $N(\mathbf{x}) = \{\mathbf{x}\}$ that local measure ρ^l_{data} becomes the point-wise error ρ_{data}. The weighting function can be chosen to follow a given distribution, e.g. uniform or Gaussian, or to be defined using the observed data. In this thesis we consider a fixed-size square function, for two cases: the uniform distribution, where weights are the unity, i.e., $w(\mathbf{x}') = 1$, and a depth-driven weighting function, where weights decay with the difference in depth relative to the window center, defined as:

$$w(\mathbf{x}') = e^{\beta |Z(\mathbf{x}) - Z(\mathbf{x}')|}, \tag{5.13}$$

reflecting the belief that close scene points are more likely to follow the same rigid motion, and avoiding the inclusion of scene points from both sides of a depth edge.

Finally, we modify the robust norm, which is able to deal with a tractable percentage of outliers, smaller than 50%, however, because total or partial occlusion of 3D surfaces, this percentage can be larger and skew motion estimation. In order to deal with of outliers brought by occlusion, the robust norm is truncated from

a given value, s_{tr}^2, as is shown in Figure 5.4, cropping these scene points from the estimation to reduce the effect of partial occlusions.

Full data term

The full data term can be written as:

$$E_D(\xi) = \sum_{\mathbf{x}} \sum_{\mathbf{x}' \in N(\mathbf{x})} w(\mathbf{x}')\, \rho_{data}\left(\mathbf{x}', \xi\right). \tag{5.14}$$

5.3 Smoothness term

Without regularization, equation (5.11) alone is not sufficient to constrain the twist motion for a given point since there is an infinite number of twists that satisfy these constraints. On the other hand, equation (5.12) allows to solve for a local motion on well textured regions but in most cases fails to capture the full motion field. We exploit the semi-rigidity of the scene to define a regularization term on the field of twist motions, which aims to favor piecewise smooth solutions. Particularly, we use TV as regularizer on this field of rigid motions for three reasons: *i*) does not penalizes discontinuities, or edges, *ii*) does not penalizes smooth functions and *iii*) preserves exactly the location of discontinuous edges. Moreover there exist efficient solvers for TV-based regularizers.

5.3.1 Regularization on the twist field

A twist motion field ξ can be decomposed into a rotational field ω, with $e^{\hat{\omega}} : \mathbf{\Omega} \rightarrow SO(3)$, and a 3D motion field $\tau : \mathbf{\Omega} \rightarrow \mathbb{R}^3$. Since they are decorrelated by nature, we regularize each field independently using a weighted TV, which allows piecewise smooth solutions while preserving motion discontinuities. Although discontinuities of these two fields are expected to coincide, a decoupled regularization scheme is preferred for three reasons: *i*) it allows to benefit from available, efficient TV regularizers, simplifying the optimization of the energy, *ii*) it yields a general framework for the twist-motion and 3D-motion representations of the scene flow and *iii*) it let to apply different TV regularization strategies to each component. Moreover, thanks to the depth data is possible to define a weighting function to preserve discontinuities.

Depth-based discontinuities preserving

In a semi-rigid scene it is expected that close 3D points move under the action of the same rigid body motion, while distant points may or may not follow the same rigid motion. However, in most cases, discontinuities of the motion field coincide

with the boundary of the observed 3D surface. This fact can be exploited using the 2D parametrization of the 3D surface given by the depth image $Z(\mathbf{x})$. Particularly, the magnitude of the gradient $|\nabla Z(\mathbf{x})|$ encodes the 3D distance between neighbor scene points and expose the edges of the surface as is shown in Figure 5.5. We define the decreasing positive function

$$c(\mathbf{x}) = e^{-\beta|\nabla Z(\mathbf{x})|^2} \tag{5.15}$$

to quantify the continuity of the surface in a given point $\mathbf{x}$, as is shown in Figure 5.5. The use of the weighting function $c(\mathbf{x})$ prevent the regularization across the discontinuities of the surface, as is presented below.

Translational field regularization

Regularization of the fields ω and τ poses different challenges. Elements of τ lie in the Euclidean space $\mathbb{R}^3$, and the problem corresponds to a vector-valued function regularization. Different vectorial TV approaches can be used to regularize τ, such as those described in (Goldluecke and Cremers, 2010). Particularly, the channel-by-channel l_1 norm has been successfully used for optical flow (Wedel *et al.*, 2009). We follow this approach to define the weighted TV of τ as:

$$\mathbf{TV}_c(\tau) = \sum_{\mathbf{x}} c(\mathbf{x}) \left\|\nabla\tau(\mathbf{x})\right\|, \tag{5.16}$$

where $\|\nabla\tau\| := |\nabla\tau_X| + |\nabla\tau_Y| + |\nabla\tau_Z|$ and $c(\mathbf{x})$ is the weighting function (5.15), which helps to preserve motion discontinuities along edges of the 3D surface. The absolute value can be replaced by the Huber norm (Werlberger *et al.*, 2009), which is given by (2.28), to reduce the staircasing effect. Efficient solvers for both norms exist and some of them are presented in (Chambolle and Pock, 2011). The staircasing effect can also be reduced if TV is replaced by the total generalized variation (TGV) Bredies *et al.* (2010), where higher order derivatives of τ are not penalized.

Rotational field regularization

Elements of ω are rotations in the Lie group $SO(3)$, embedded in $\mathbb{R}^3$ through the exponential map (2.7), and the regularization has to be done on this manifold. In order to apply a TV regularization, a notion of variation should be used for elements of $SO(3)$, which is not a vector space. Given two rotations $e^{\hat{\omega}_1}, e^{\hat{\omega}_2} \in SO(3)$, the residual rotation can be defined as $e^{\hat{\omega}_2}e^{-\hat{\omega}_1}$, representing the rotation required to go from $e^{\hat{\omega}_1}$ to $e^{\hat{\omega}_2}$ in $SO(3)$. The product in logarithmic coordinates can be expressed as $e^{\hat{\omega}_1}e^{\hat{\omega}_2} = e^{\mu(\hat{\omega}_1,\hat{\omega}_2)}$, where the mapping μ can be expanded in a Taylor series

Fig. 5.5: *Depth-based discontinuities preserving. From left to right: (a) depth image, $Z(x)$, (b) magnitude of the gradient, $|\nabla Z(x)|$, and (c) depth-based weighting function, $c(\mathbf{x})$.*

around the identity, using the Baker-Campbell-Hausdorff formula:

$$\mu(\hat{\omega}_1, \hat{\omega}_2) = \hat{\omega}_1 + \hat{\omega}_2 + \frac{1}{2}[\hat{\omega}_1, \hat{\omega}_2] + O(|(\hat{\omega}_1, \hat{\omega}_2)|^3), \tag{5.17}$$

where $[\cdot, \cdot]$ is the Lie bracket in $so(3)$. Close to the identity, equation (5.17) is well approximated by its first-order terms, yielding to $e^{\hat{\omega_2}} e^{-\hat{\omega}_1} \approx \hat{\omega}_2 - \hat{\omega}_1$. Therefore, for small rotations the variation measure can be defined as the matrix subtraction in $so(3)$, or equivalently, as a vector difference for the embedding in $\mathbb{R}^3$. Accordingly, with ∇ being the image gradient operator, the derivative image matrix, $\mathbf{D}\omega := (\nabla\omega_X, \nabla\omega_Z, \nabla\omega_Z)^T : \mathbf{\Omega} \to \mathbb{R}^{3\times 2}$, approximates the horizontal and vertical pointwise variations of ω on the image. Following (Goldluecke and Cremers, 2010), we define the TV as the sum over the largest singular value σ_1 of the derivative matrix:

$$\mathbf{TV}_\sigma(\omega) = \sum_{\mathbf{x}} c(\mathbf{x})\, \sigma_1(\mathbf{D}\omega(\mathbf{x})) \ . \tag{5.18}$$

This TV approach supports a common edge direction for the three components, which is a desirable property for regularization of the field of rotations. Moreover, deviations are less penalized with respect to other measures, e.g., the Frobenius norm (Rosman *et al.*, 2012), and efficient solvers are available. However, this TV definition approximates the real structure of the manifold, yielding biased measures when it is far from the identity. The more the rotation is away from the identity, the more its variations in $SO(3)$ are penalized as is shown hereinafter. Given two rotations $\omega_1 = \theta_1 \overleftarrow{\omega_1}$ and $\omega_2 = \theta_2 \overleftarrow{\omega_2}$, with θ the angle and $\overleftarrow{\omega}$ the unitary axis vector

of the rotation, and writing $\theta_2 = \theta_1 + \delta\theta$, leads to $\omega_2 - \omega_1 = \theta_1(\overleftarrow{\omega_2} - \overleftarrow{\omega_1}) - \delta\theta\overleftarrow{\omega_2}$. This linearly dependent penalization usually is not a problem, since large rotations imply larger motion on the image. Thus, a stronger regularization can be reasonable. Moreover, large rotation caused by a global motion of the scene or the camera can be optimized separately and compensated, as we present in Section 5.6. Optionally, this over-penalization can be removed by expressing each rotation as $\omega = \theta\overleftarrow{\omega}$ and applying a vectorial TV on $\overleftarrow{\omega}$ and a scalar TV on θ. This alternative definition does not increase the computational cost much, since the solver for (5.18) computes the norm of every element of ω and the field generated by θ is efficiently regularized using a scalar TV solver.

Full regularization term

The regularization on the twist field is formulated as:

$$E_S(\xi) = \mathbf{TV}_c(\tau) + \mathbf{TV}_\sigma(\omega). \tag{5.19}$$

5.3.2 Regularization on the 3D motion field

When a rigid surface moves in the scene, the 3D distance between any two of its points remains constant. Therefore, it is intuitive to expect that close points perform the same, or a very similar, 3D motion, and encourage piecewise smooth solutions through the regularization. This assumption has been successfully used in the regularization of the optical flow. However, although 3D distances remain unchanged, 3D motions performed for points of the moving rigid surface are not the same in general. Using the small-rotation model for a rigid motion, given by Equation (2.16), the scene flow $\mathbf{v}$ induced on a 3D point $\mathbf{X}$, is given by:

$$\begin{pmatrix} v_X \\ v_Y \\ v_Z \end{pmatrix} = \begin{pmatrix} 0 & -Z & Y \\ Z & 0 & -X \\ -Y & X & 0 \end{pmatrix} \begin{pmatrix} \omega_X \\ \omega_Y \\ \omega_Z \end{pmatrix} + \begin{pmatrix} t_X \\ t_Y \\ t_Z \end{pmatrix}, \tag{5.20}$$

from which is clear that if the rotational component is not zero, the scene flow is a function of the 3D point, and so that, it differs from one point to another on the rigid surface. Only a zero rotation component of the rigid motion, i.e., a pure translational motion, guarantees a constant scene flow. Using the 3D-motion representation of the scene flow is not possible to directly exploit the piecewise rigidity of the scene. However, close 3D points that follows the same rigid motion, are expected to present similar 3D motions and piecewise smoothness of the 3D motion field is a valid approximation. Therefore, we propose a regularization approach using this piecewise smoothness assumption combined with a rigid motion prior (Vogel *et al.*, 2011), which encourages a local rigidity without an explicit

computation of the rigid motion field. This way local and piece-wise rigidity are combined into the regularization of the 3D motion field.

TV-based regularization on the 3D motion field

In order to favor smooth piecewise solution of the scene flow, a weighted TV is applied. The regularization is done channel by channel according to:

$$\mathbf{TV}_c(\mathbf{v}) = \sum_{\mathbf{x}} c(\mathbf{x}) \left\| \nabla \mathbf{v}(\mathbf{x}) \right\|, \tag{5.21}$$

with $\|\nabla \mathbf{v}\| := |\nabla v_X| + |\nabla v_Y| + |\nabla v_Z|$ and $c(\mathbf{x})$ the weighting function (5.15).

Rigid-motion prior on the 3D motion field

Aiming to exploit the semi-rigidity of the scene, the 3D motion field is encouraged to be locally consistent with a rigid motion. Let $\mathbf{NR}(\mathbf{v})$ be the nonrigid residual functional, defined as:

$$\mathbf{NR}\left(\mathbf{v}\right) = \sum_{\mathbf{x}} \Psi\left(\left|\mathbf{Rig}_{\mathbf{v}}^{N}(\mathbf{x}) - \mathbf{v}(\mathbf{x})\right|^2\right) \tag{5.22}$$

where $\mathbf{Rig}_{\mathbf{v}}^{N}(\mathbf{x})$ is the local $N \times N$ projection of the scene flow $\mathbf{v}$ onto the closest rigid motion subspace. The functional $\mathbf{NR}\left(\mathbf{v}\right)$ measures the total non-local rigidity of the scene flow and it is provided with the robust norm $\Psi\left(s^2\right) = \sqrt{s^2 + 0.001^2}$, to deal with outliers. The definition of the local rigid projection $\mathbf{Rig}_{\mathbf{v}}^{N}(\mathbf{x})$ is based on the local fitting of a rigid motion $(\omega, \mathbf{t})$, using the small-rotation approximation (5.20). With $N_{(\mathbf{x})}$ being a $N \times N$ neighborhood centering at $\mathbf{x}$, the local rigid motion $(\omega, \mathbf{t})$ is defined via the following weighted least-squares problem:

$$\min_{(\omega,\mathbf{t})} \sum_{\mathbf{x}' \in N(\mathbf{x})} w(\mathbf{x}, \mathbf{x}') \left| \left([\,\mathbf{X}']_\times \,|\, \mathbf{I}_{3\times 3} \right) \begin{pmatrix} \omega \\ \mathbf{t} \end{pmatrix} - \mathbf{v}(\mathbf{x}') \right|^2, \tag{5.23}$$

with $\mathbf{X}' = \pi^{-1}(\mathbf{x}', Z(\mathbf{x}'))$ and $[\cdot]_\times$ the cross product matrix. The weighting function $w_{\mathbf{x}}(\mathbf{x}, \mathbf{x}')$ aim to favor points belonging to the same surface of $\mathbf{x}$. For this purpose, each pixel $\mathbf{x}'$ is weighted by $e^{-kd(\mathbf{x},\mathbf{x}')^2}$, where $d(\mathbf{x}, \mathbf{x}')$ is the 3D Euclidean distance between $\mathbf{x}$ and $\mathbf{x}'$, and k an adjustable parameter. We follow the solution of problem (5.23) presented by Vogel *et al.* (2011). Let $M \in \mathbb{R}^{3N^2 \times 6}$ and $P \in \mathbb{R}^{3N^2 \times 1}$ be the concatenation of matrices $([\,\mathbf{X}']_\times \,|\, \mathbf{I}_{3\times 3}) \in \mathbb{R}^{3\times 6}$ and scene flow vectors $\mathbf{v}(\mathbf{x}') \in \mathbb{R}^3$, respectively, whose elements are taken from $N_{\mathbf{x}}$. Let $\mu \in \mathbb{R}^{3N^2 \times 3N^2}$ be the diagonal matrix, whose elements correspond to the weights given

by $w_{\mathbf{x}}(\mathbf{x}, \mathbf{x}')$. Using this notation, the rigid motion that solves (5.23) is given by:

$$\begin{pmatrix} \omega \\ \mathbf{t} \end{pmatrix} = (M^T \mu M)^{-1} M^T \mu P \,, \tag{5.24}$$

therefor, the projection of $\mathbf{v}(\mathbf{x})$ onto the local rigid motion subspace is given by:

$$\mathbf{Rig}_{\mathbf{v}}^{N}(\mathbf{x}) = ([\,\mathbf{X}]_{\times} \,|\, \mathbf{I}_{3\times 3}\,)\,(M^T \mu M)^{-1} M^T \mu P \,. \tag{5.25}$$

The rigid projection requires the inversion of the matrix $(M^T \mu M) \in \mathbb{R}^{6\times 6}$, which can be done efficiently. Although equation (5.25) allows the projection of a local scene flow without the explicit computation of the rigid motion, due to the linear approximation it assumes a small rotation. It is also possible to iteratively compute the full rigid motion that best fit the local scene flow, in which case, is required to explicitly compute $(\omega, \mathbf{t})$.

Full regularization term

The regularization term of the scene flow is defined as:

$$E_S\,(\mathbf{v}) = \mathbf{TV}_c(\mathbf{v}) + \zeta\,\mathbf{NR}(\mathbf{v}), \tag{5.26}$$

where ζ balances the total variation and the locally-rigid terms.

5.4 Optimization of the energy

The minimization of the proposed energy (5.2) is a challenge due to the l_1-norm used in the regularization term, which is non-differentiable. However, it is possible to introduce an auxiliary variable (Wang *et al.*, 2008) to decompose the optimization into two simpler problems. In particular, the auxiliary twist field χ is used to reformulate the minimization problem as follows:

$$\min_{\xi,\chi}\; E_D(\xi) + \frac{1}{2\kappa}\sum_{\mathbf{x}} |\xi(\mathbf{x}) - \chi(\mathbf{x})|^2 + \alpha E_S(\chi) \tag{5.27}$$

where κ is a small numerical variable. Note that the linking term between ξ and χ is the distance on the tangent space at the identity in $SE(3)$. The solution of (5.27) converges to that of (5.2) as $\kappa \to 0$. Minimization of this energy is performed by alternating the following two optimization problems:

i. For fixed χ, estimate ξ that minimizes:

$$\sum_{\mathbf{x}} \rho^{l}_{data}\left(\mathbf{x}, \xi(\mathbf{x})\right) + \frac{1}{2\kappa}\left|\xi(\mathbf{x}) - \chi(\mathbf{x})\right|^2. \tag{5.28}$$

ii. For fixed $\xi = (\omega, \tau)$, compute $\chi = (\varpi, \pi)$ that minimizes:

$$\left\{\mathbf{TV}_c(\pi) + \frac{\eta}{2}\sum_{\mathbf{x}}\left|\pi(\mathbf{x}) - \tau(\mathbf{x})\right|^2\right\} + \left\{\mathbf{TV}_\sigma(\varpi) + \frac{\eta}{2}\sum_{\mathbf{x}}\left|\varpi(\mathbf{x}) - \omega(\mathbf{x})\right|^2\right\}, \tag{5.29}$$

where $\eta = (\kappa\alpha)^{-1}$. Below we present how to solve for data-based and smoothness-based problems, which are defined as the minimization of (5.28) and (5.29), respectively. Afterwards, the minimization of the scene flow energy for the 3D-motion representation is presented.

5.4.1 Data-based energy minimization

The optimization of (5.28) can be done pointwise by minimizing:

$$\frac{1}{2\kappa}\left|\xi - \chi\right|^2 + \sum_{\mathbf{x}\in N(\mathbf{x}')} \Psi\left(\rho_I^2\left(\mathbf{x}, \xi\right) + \gamma\rho_g^2\left(\mathbf{x}, \xi\right)\right) + \lambda\Psi\left(\rho_Z^2\left(\mathbf{x}, \xi\right)\right). \tag{5.30}$$

This minimization corresponds to a nonlinear optimization problem, which can be solved by IRLS (Green, 1984) with a Gauss-Newton numerical algorithm. For this purpose, the energy is linearized at a given estimate ξ, to calculate a rigid motion increment, $\Delta\xi$, on the tangent space to the manifold. This procedure is applied iteratively and the twist motion updating is done via $\xi \leftarrow \log(e^{\hat{\Delta\xi}}e^{\hat{\xi}})$.

Linearization of the data-based energy

Energy (5.30) can be linearized around the current estimate ξ, using a first-order Taylor series expansion. Particularly, for the brightness residual we have:

$$\rho_I\left(\mathbf{x}, \log(e^{\hat{\Delta\xi}}e^{\hat{\xi}})\right) \approx I_2(\mathbf{x}_\xi) + \mathbf{I}_\mathbf{x}\frac{\partial\mathbf{W}}{\partial\xi}\Delta\xi - I_1\left(\mathbf{x}\right), \tag{5.31}$$

where $\mathbf{I}_\mathbf{x} = (\partial I_2/\partial x, \partial I_2/\partial y)$ is the image gradient and $\partial\mathbf{W}/\partial\xi$ is the Jacobian $\mathbf{J}$ of the warp, which is given by (4.10), and both evaluated at the current estimate of the scene point position, $\mathbf{x}_\xi = \mathbf{W}(\mathbf{x}, \xi)$. Therefore, the residual becomes:

$$\rho_I\left(\mathbf{x}, \log(e^{\hat{\Delta\xi}}e^{\hat{\xi}})\right) \approx \rho_I\left(\mathbf{x}, \xi\right) + \mathbf{I}_\mathbf{x}\mathbf{J}\Delta\xi, \tag{5.32}$$

and, similarly, for the gradient residual we have:

$$\rho_g\left(\mathbf{x}, \log(e^{\Delta\hat{\xi}}e^{\hat{\xi}})\right) \approx \rho_g\left(\mathbf{x}, \xi\right) + \mathbf{I}_{\mathbf{x}}^g \mathbf{J}\Delta\xi. \tag{5.33}$$

Using the first-order expansion for the depth residual yields:

$$\rho_Z\left(\mathbf{x}, \log(e^{\Delta\hat{\xi}}e^{\hat{\xi}})\right) \approx Z_2(\mathbf{x}_\xi) + \mathbf{Z}_{\mathbf{x}}\mathbf{J}\Delta\xi - \left(Z_1\left(\mathbf{x}\right) + \delta_Z\left(\mathbf{x}, \log(e^{\Delta\hat{\xi}}e^{\hat{\xi}})\right)\right), \tag{5.34}$$

where $\mathbf{Z}_{\mathbf{x}} = (\partial Z_2/\partial x, \partial Z_2/\partial y)$ is the depth image gradient evaluated at $\mathbf{x}_\xi$. The last term, δ_Z, is the variation in depth which can be linearized by approximating the 3D motion, $\delta_{\mathbf{3D}}$, induced by $\Delta\xi$, as follows:

$$\begin{aligned}\delta_{\mathbf{3D}}(\mathbf{x}, \log(e^{\Delta\hat{\xi}}e^{\hat{\xi}})) &= \left(e^{\Delta\hat{\xi}}e^{\hat{\xi}} - \mathbf{I}\right)\tilde{\mathbf{X}} \approx \left((\mathbf{I} + \Delta\hat{\xi})e^{\hat{\xi}} - \mathbf{I}\right)\tilde{\mathbf{X}} \\ &\approx \left(e^{\hat{\xi}} - \mathbf{I}\right)\tilde{\mathbf{X}} + \Delta\hat{\xi}\left(e^{\hat{\xi}}\tilde{\mathbf{X}}\right) = \delta_{\mathbf{3D}}(\mathbf{x}, \xi) + \Delta\hat{\xi}\tilde{\mathbf{X}}_\xi.\end{aligned} \tag{5.35}$$

Therefore, the depth residual can be written as:

$$\rho_Z\left(\mathbf{x}, \log(e^{\Delta\hat{\xi}}e^{\hat{\xi}})\right) \approx \rho_Z\left(\mathbf{x}, \xi\right) + \left(\mathbf{Z}_{\mathbf{x}}\mathbf{J} - \mathbf{K}\right)\Delta\xi, \tag{5.36}$$

where the 1×6 vector $\mathbf{K}$ is defined as $\mathbf{K} = \mathbf{D}\left([\mathbf{X}_\xi]_\times | \mathbf{I}_{3\times 3}\right)$, with $\mathbf{D} = (0, 0, 1)$ isolating the third component and $[\cdot]_\times$ denoting the cross product matrix. Finally, the linearized version of the data term can be expressed as:

$$\begin{aligned}\frac{1}{2\kappa}\left|\log(e^{\Delta\hat{\xi}}e^{\hat{\xi}}) - \chi\right|^2 + \sum_{\mathbf{x}\in N(\mathbf{x}')} \Psi\left(\left|\rho_I + \mathbf{I}_{\mathbf{x}}\mathbf{J}\Delta\xi\right|^2 + \gamma\left|\rho_g + \mathbf{I}_{\mathbf{x}}^g\mathbf{J}\Delta\xi\right|^2\right) \\ + \lambda\Psi\left(\left|\rho_Z + \left(\mathbf{Z}_{\mathbf{x}}\mathbf{J} - \mathbf{K}\right)\Delta\xi\right|^2\right),\end{aligned} \tag{5.37}$$

where the parameters $(\mathbf{x}, \xi)$ are considered implicit.

Twist motion increment

Taking the partial derivative of (5.37) with respect to $\Delta\xi$ and setting it to zero, the increment $\Delta\xi$ can be computed as:

$$\Delta\xi = -\mathbf{H}^{-1}\mathbf{g}, \tag{5.38}$$

where the 6×1 *gradient vector* $\mathbf{g}$ is given by:

$$\begin{aligned}\mathbf{g} = \sum_{\mathbf{x}\in N(\mathbf{x}')} \left\{\Psi'\left(\rho_I^2 + \gamma\rho_g^2\right)\left[(\mathbf{I}_{\mathbf{x}}\mathbf{J})^T\rho_I + \gamma(\mathbf{I}_{\mathbf{x}}^g\mathbf{J})^T\rho_g\right]\right. \\ \left. + \lambda\,\Psi'\left(\rho_Z^2\right)\left(\mathbf{Z}_{\mathbf{x}}\mathbf{J} - \mathbf{K}\right)^T\rho_Z\right\} + \frac{1}{\kappa}\log\left((e^{\hat{\xi}})^{-1}e^{\hat{\chi}}\right),\end{aligned} \tag{5.39}$$

with Ψ' the derivative of the robust norm, which is always evaluated at the current estimate ξ. The 6×6 matrix $\mathbf{H}$ is the *Gauss-Newton approximation of the Hessian matrix*, which is given by:

$$\mathbf{H} = \sum_{\mathbf{x} \in N(\mathbf{x}')} \Psi'\left(\rho_I^2 + \gamma \rho_g^2\right) \left[(\mathbf{I_x J})^T (\mathbf{I_x J}) + \gamma \, (\mathbf{I}_\mathbf{x}^g \mathbf{J})^T (\mathbf{I}_\mathbf{x}^g \mathbf{J}) \right] + \lambda \, \Psi'\left(\rho_Z^2\right) (\mathbf{Z_x J} - \mathbf{K})^T (\mathbf{Z_x J} - \mathbf{K}) + \frac{1}{2\kappa} \mathbf{I}_{6\times 6}. \quad (5.40)$$

A reliable computation of (5.38) requires that the Hessian matrix $\mathbf{H}$ is invertible and well conditioned. The construction of $\mathbf{H}$ is done as a weighted linear combination of 6×6 matrices, which are computed using gradient images $\mathbf{I_x}$, $\mathbf{I}_\mathbf{x}^g$ and $\mathbf{Z_x}$, and the Jacobian $\mathbf{J}$. The main drawback of local methods following the Lucas-Kanade framework (Baker and Matthews, 2004), occurs on regions presenting vanishing gradients, in both intensity and depth images for RGBD data. In such that case, the Hessian matrix $\mathbf{H}$ is non-invertible and/or ill-conditioned, and so that, being unable to perform dense motion estimation. However, in our approach, the regularization applied includes the auxiliary field χ, which provides an interesting stability to the optimization procedure. The identity matrix $\mathbf{I}_{6\times 6}$ allows to smoothly switch between the Gauss-Newton algorithm and a gradient descent algorithm. On one hand, if the gradients vanish, $\mathbf{H}$ tends to a constant diagonal matrix and the twist increment equation (5.38) leans to a gradient descent updating where the initial estimation is favored. On the other hand, if the gradient information is meaningful, $\mathbf{H}^{-1}$ guides the twist motion increment according to the gradient images and the Jacobian, in a Gauss-Newton method. This way, equation (5.38) avoids to estimate motions on flat regions of the image, giving to the smoothness procedure the task of completing the estimation. Unlike the 2D motion on the image plane, depth changes can always be computed as is demonstrated in Section 5.4.3, where a close look to the Hessian Matrix is done for the 3D-motion representation case.

5.4.2 Smoothness-based energy minimization

Each side of the energy (5.29) can be seen as vectorial image denoising problem with a TV-L^2 model. The left side corresponds to a channel-by-channel TV. This regularization is done independently per channel and coincides with the ROF-model (Rudin *et al.*, 1992). This problem is solved using the Chambolle's projection algorithm (Chambolle, 2005), which is based on gradient descent and re-projection on the dual-ROF model. The right side is a vectorial TV regularization proposed by Goldluecke and Cremers (2010). This problem can be solved using the Bermudez-Moreno algorithm (Bermudez and Moreno, 1981; Goldluecke *et al.*, 2012), which is a generalization of the Chambolle's projection algorithm (Aujol, 2009). Below we describe the solution of both TV-based problems.

Channel-by-channel TV

Given a fixed 3D motion field $\tau = (\tau_X, \tau_Y, \tau_Z)$, the goal is to compute $\pi = (\pi_X, \pi_Y, \pi_Z)$ that minimizes the left side of the energy (5.29). For every channel $i = \{X, Y, Z\}$, this problem corresponds to the following weighted formulation of the ROF image denoising model:

$$\sum_{\mathbf{x}} c(\mathbf{x}) \, |\nabla \pi_i(\mathbf{x})| + \frac{\eta}{2} |\pi_i(\mathbf{x}) - \tau_i(\mathbf{x})|^2. \tag{5.41}$$

An efficient solution, using the dual ROF-model, is given by the Chambolle's projection algorithm (Chambolle, 2005) according to:

$$\pi_i(\mathbf{x}) = \tau_i(\mathbf{x}) + \frac{c(\mathbf{x})}{\eta} \nabla \cdot \mathbf{p}(\mathbf{x}) \,, \tag{5.42}$$

where $\mathbf{p} = (p_1, p_2) : \mathbf{\Omega} \to \mathbb{R}^2$ is the solution of the dual problem, which is defined (almost everywhere) as the unite vector $\nabla \pi_i / |\nabla \pi_i|$. Setting $\mathbf{p}^0 = \mathbf{0}$, $\pi_i^0 = \tau_i$ and choosing $\nu \leq 1/4$, the dual variable $\mathbf{p}$ can be computed recursively following:

$$\mathbf{p}^{n+1}(\mathbf{x}) = \Pi_S \left(\mathbf{p}^n(\mathbf{x}) + (\eta\nu) \, \nabla \pi_i^n(\mathbf{x}) \right), \tag{5.43}$$

with the projection Π_S defined as:

$$\Pi_S \left(\mathbf{p}(\mathbf{x}) \right) = \frac{\mathbf{p}(\mathbf{x})}{\max \{1, |\mathbf{p}(\mathbf{x})|\}}. \tag{5.44}$$

Vectorial TV

Given a rotational field $\omega = (\omega_X, \omega_Y, \omega_Z)$ one wants to solve for $\varpi = (\varpi_X, \varpi_Y, \varpi_Z)$ that minimizes:

$$\sum_{\mathbf{x}} c(\mathbf{x}) \, \sigma_1(\mathrm{D}\omega(\mathbf{x})) + \frac{\eta}{2} |\varpi(\mathbf{x}) - \omega(\mathbf{x})|^2 \,, \tag{5.45}$$

This problem is solved using the Bermudez-Moreno's framework (Aujol, 2009). This is a general projection-based algorithm that can be used to solve the dual problem of a vectorial TV image denoising, as is presented by Goldluecke *et al.* (2012). Using the Bermudez-Moreno's algorithm the minimizer of (5.45) is given by:

$$\varpi(\mathbf{x}) = \omega(\mathbf{x}) + \frac{c(\mathbf{x})}{\eta} \operatorname{Div} \mathbf{P}(\mathbf{x}) \,, \tag{5.46}$$

where $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) : \mathbf{\Omega} \to \mathbb{R}^{3\times 2}$ is the solution of the dual problem, and $\operatorname{Div} \mathbf{P}$, the *divergence matrix* defined as $\operatorname{Div} \mathbf{P} := (\nabla \cdot \mathbf{p}_1, \nabla \cdot \mathbf{p}_2, \nabla \cdot \mathbf{p}_3)^T : \mathbf{\Omega} \to \mathbb{R}^3$. Setting $\mathbf{P}^0 = \mathbf{0}$, $\varpi^0 = \omega$ and choosing $\nu \leq 1/4$, the dual variable $\mathbf{P}$ can be computed

recursively following:

$$\mathbf{P}^{n+1}(\mathbf{x}) = \Pi_K \left(\mathbf{P}^n(\mathbf{x}) + (\eta\nu)\, \mathrm{D}\varpi^n(\mathbf{x}) \right), \tag{5.47}$$

with $\mathrm{D}\omega := (\nabla\omega_X, \nabla\omega_Z, \nabla\omega_Z)^T : \mathbf{\Omega} \to \mathbb{R}^3$, the derivative matrix of ω, and the projection, Π_K, defined as:

$$\Pi_K \left(\mathbf{P}(\mathbf{x}) \right) = \mathbf{P}\mathbf{V}\mathbf{\Sigma}^{+}\mathbf{\Sigma}_{l_1}\mathbf{V}^T. \tag{5.48}$$

Being $\mathbf{P} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$ the singular value decomposition of $\mathbf{P}$, and $\mathbf{\Sigma}$, the diagonal matrix containing the singular values $\{\sigma_1, \sigma_2\}$, matrix $\mathbf{\Sigma}^{+}$ denotes the pseudo inverse of $\mathbf{\Sigma}$ and matrix $\mathbf{\Sigma}_{l_1}$ the projection of the elements of $\mathbf{\Sigma}$ onto the l_1-unit ball.

5.4.3 Minimization on the 3D motion field

Using the 3D-motion representation of the scene flow, the problem is formulated as the minimization of the following energy:

$$E(\mathbf{v}) = E_D(\mathbf{v}) + \alpha \left\{ \mathbf{TV}_c(\mathbf{v}) + \lambda\, \mathbf{NR}(\mathbf{v}) \right\}, \tag{5.49}$$

where the data term, E_D, is given by (5.14), with $\xi = (\mathbf{0}, \mathbf{v})$, i.e., a zero rotational component, and the smoothness term is formed by the total variation and nonrigid terms. The minimization of the energy (5.49) is done by introducing two auxiliary motion fields to decompose the optimization into three simpler problems. Let $\mathbf{u}$ and $\mathbf{w}$ be 3D motion fields, then the scene flow $\mathbf{v}$ is solved to minimize:

$$E_D(\mathbf{v}) + \frac{1}{2\kappa} \sum_{\mathbf{x}} |\mathbf{v}(\mathbf{x}) - \mathbf{u}(\mathbf{x})|^2 + \alpha \mathbf{TV}_c(\mathbf{u}) + \frac{1}{2\kappa} \sum_{\mathbf{x}} |\mathbf{v}(\mathbf{x}) - \mathbf{w}(\mathbf{x})|^2 + \alpha\lambda\, \mathbf{NR}(\mathbf{w}). \tag{5.50}$$

The minimum of (5.50) converges to the solution of (5.49) as $\kappa \to 0$, and can be computed by alternating the three following optimization problems:

i. For fixed $\mathbf{u}$ and $\mathbf{w}$, estimate $\mathbf{v}$ that minimizes:

$$\sum_{\mathbf{x}} \rho^l_{data}\left(\mathbf{x}, \mathbf{v}(\mathbf{x})\right) + \frac{1}{2\kappa} \left\{ |\mathbf{v}(\mathbf{x}) - \mathbf{u}(\mathbf{x})|^2 + |\mathbf{v}(\mathbf{x}) - \mathbf{w}(\mathbf{x})|^2 \right\}. \tag{5.51}$$

ii. For fixed $\mathbf{v}$, compute $\mathbf{u}$ that minimizes:

$$\mathbf{TV}_c(\mathbf{u}) + \frac{\eta}{2} \sum_{\mathbf{x}} |\mathbf{u}(\mathbf{x}) - \mathbf{v}(\mathbf{x})|^2, \tag{5.52}$$

where $\eta = (\kappa\alpha)^{-1}$.

iii. For fixed $\mathbf{v}$, compute $\mathbf{w}$ that minimizes:

$$\mathbf{NR}(\mathbf{w}) + \frac{\eta}{2}\sum_{\mathbf{x}} |\mathbf{w}(\mathbf{x}) - \mathbf{v}(\mathbf{x})|^2, \tag{5.53}$$

with $\eta = (\kappa\alpha\lambda)^{-1}$.

Problem **ii**. is solved using the channel-by-channel TV approach presented in Section 5.4.2. The minimization of problem **i**., or data-based energy, and of problem **iii**., or nonrigid-based energy, is presented below.

Data-based energy minimization

The optimization of (5.51) can be done pointwise by minimizing:

$$\frac{1}{2\kappa}\left\{|\mathbf{v}-\mathbf{u}|^2 + |\mathbf{v}-\mathbf{w}|^2\right\} + \sum_{\mathbf{x}\in N(\mathbf{x}')} \Psi\left(\rho_I^2(\mathbf{x},\mathbf{v}) + \gamma\rho_g^2(\mathbf{x},\mathbf{v})\right) + \lambda\Psi\left(\rho_Z^2(\mathbf{x},\mathbf{v})\right). \tag{5.54}$$

This problem can be solved using IRLS with a Gauss-Newton algorithm, where the scene flow increment, $\Delta\mathbf{v}$, is iteratively calculated by linearizing the energy at the current estimate $\mathbf{v}$. Thus, the scene flow solution is updating via $\mathbf{v} \leftarrow \mathbf{v} + \Delta\mathbf{v}$. Following (5.37), energy (5.54) is linearized around the current estimate $\mathbf{v}$ as follows:

$$\frac{1}{2\kappa}\left\{|\mathbf{v}-\mathbf{u}|^2 + |\mathbf{v}-\mathbf{w}|^2\right\} + \sum_{\mathbf{x}\in N(\mathbf{x}')} \Psi\left(|\rho_I + (\mathbf{I}_\mathbf{x}\mathbf{J})\Delta\mathbf{v}|^2 + \gamma\,|\rho_g + (\mathbf{I}^g_\mathbf{x}\mathbf{J})\Delta\mathbf{v}|^2\right) + \lambda\Psi\left(|\rho_Z + (\mathbf{Z}_\mathbf{x}\mathbf{J} - \mathbf{D})\,\Delta\mathbf{v}|^2\right), \tag{5.55}$$

where the Jacobian $\mathbf{J}$ of the warp, is given by Equation (4.16) and the 3D vector $\mathbf{D} = (0,0,1)$ isolates the Z-component. Taking the partial derivative of (5.55) with respect to $\Delta\mathbf{v}$, and setting it to zero, the increment $\Delta\mathbf{v}$ can be computed as:

$$\Delta\mathbf{v} = -\mathbf{H}^{-1}\mathbf{g}, \tag{5.56}$$

where $\mathbf{g}$ is the 3×1 *gradient vector* defined as:

$$\mathbf{g} = \sum_{\mathbf{x}\in N(\mathbf{x}')} \left\{\Psi'\left(\rho_I^2 + \gamma\rho_g^2\right)\left[(\mathbf{I}_\mathbf{x}\mathbf{J})^T\rho_I + \gamma(\mathbf{I}^g_\mathbf{x}\mathbf{J})^T\rho_g\right] + \lambda\,\Psi'\left(\rho_Z^2\right)(\mathbf{Z}_\mathbf{x}\mathbf{J} - \mathbf{D})^T\rho_Z\right\} + \frac{2}{\kappa}\left(\mathbf{v} - \frac{\mathbf{u}+\mathbf{w}}{2}\right), \tag{5.57}$$

with the derivative of the robust norm Ψ' evaluated at the current estimate $\mathbf{v}$, and the 3×3 approximation of the Hessian matrix $\mathbf{H}$ given by:

$$\mathbf{H} = \sum_{\mathbf{x} \in N(\mathbf{x}')} \Psi'\left(\rho_I^2 + \gamma \rho_g^2\right) \left[\left(\mathbf{I}_\mathbf{x}\mathbf{J}\right)^T \left(\mathbf{I}_\mathbf{x}\mathbf{J}\right) + \gamma \left(\mathbf{I}_\mathbf{x}^g\mathbf{J}\right)^T \left(\mathbf{I}_\mathbf{x}^g\mathbf{J}\right) \right] + \lambda \Psi'\left(\rho_Z^2\right) \left(\mathbf{Z}_\mathbf{x}\mathbf{J} - \mathbf{D}\right)^T \left(\mathbf{Z}_\mathbf{x}\mathbf{J} - \mathbf{D}\right) + \frac{1}{\kappa}\mathbf{I}_{3\times 3}. \quad (5.58)$$

The construction of $\mathbf{H}$ is a weighted linear combination of 3×3 matrices, which are computed from gradient images $\mathbf{I}_\mathbf{x}$, $\mathbf{I}_\mathbf{x}^g$ and $\mathbf{Z}_\mathbf{x}$, and the Jacobian, $\mathbf{J}$, of the warp. For this reason, the gradient information determines the ability to estimate motion in the image domain for a given region. For instance, each of the matrices which depends on the intensity gradient, can be written as:

$$\left(\mathbf{I}_\mathbf{x}\mathbf{J}\right)^T \left(\mathbf{I}_\mathbf{x}\mathbf{J}\right) = \mathbf{J}^T \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \mathbf{J} = \frac{1}{Z(\mathbf{x})^2} \begin{pmatrix} I_x^2 & I_x I_y & I_x I_\Sigma \\ I_x I_y & I_y^2 & I_y I_\Sigma \\ I_x I_\Sigma & I_y I_\Sigma & I_\Sigma^2 \end{pmatrix} \quad (5.59)$$

with $I_\Sigma = -\left(x I_x + y I_y\right)$, combining horizontal and vertical components of the gradient as a function of the pixel location. The 2×2 inner matrix in (5.59), $\mathbf{I}_\mathbf{x}^T \mathbf{I}_\mathbf{x}$, characterizes the gradient information pointwise, and when summed with all matrices in the neighborhood, produces the *covariance matrix of the gradient*, which has been used to determine regions that can be tracked well (Shi and Tomasi, 1994). The full 3×3 matrix takes into account the projective transformation depending on the pixel location, and the inverse square of its depth, which weighs the contribution of every matrix. If the point-wise gradient vanishes or $\Psi' \to 0$, due to the presence of an outlier, there is a null contribution of this pixel into the overall construction of $\mathbf{H}$. Gradient and depth images also contribute in the solution, particularly, the gradient of the depth image brings a decorrelated information, capturing the texture of the 3D surface which benefits the characterization of the region. The matrices depending on the depth gradient take the following form:

$$\left(\mathbf{Z}_\mathbf{x}\mathbf{J}\right)^T \left(\mathbf{I}_\mathbf{x}\mathbf{J}\right) = \frac{1}{Z(\mathbf{x})^2} \begin{pmatrix} Z_x^2 & Z_x Z_y & Z_x\left(Z_\Sigma - 1\right) \\ Z_x Z_y & Z_y^2 & Z_y\left(Z_\Sigma - 1\right) \\ Z_x\left(Z_\Sigma - 1\right) & Z_y\left(Z_\Sigma - 1\right) & \left(Z_\Sigma - 1\right)^2 \end{pmatrix} \quad (5.60)$$

with $Z_\Sigma = -\left(x Z_x + y Z_y\right)$. This way gradient information from intensity and depth images complement each other, allowing motion estimation on regions where the use alone of each source of data cannot. However, when both intensity and depth gradients vanish, it is not possible to estimate motion in the image since $\mathbf{I}_\mathbf{x}\mathbf{J}$, $\mathbf{I}_\mathbf{x}^g\mathbf{J}$ and $\mathbf{Z}_\mathbf{x}\mathbf{J}$ in equation (5.56) go to zero. Observe that even on this untextured region the matrix $\mathbf{H}$ is still invertible thanks to the regularization. This is an important

properties because it prevents the estimation of the 2D motion on regions with no enough gradient information, conversely, it is still able to compute the Z-component of the motion. Observe that when gradients vanish the 2D motion estimation is done in the regularization step, which corrects and completes the estimation using the global information. However, at the same time, if intensity and depth gradients vanish the increment equation (5.56) becomes:

$$\Delta\mathbf{v} = \mathbf{H}^{-1}\left[\sum_{\mathbf{x}\in N(\mathbf{x}')}\left\{\lambda\,\Psi'\left(\rho_Z^2\right)\mathbf{D}^T\rho_Z\right\} + \frac{2}{\kappa}\left(\mathbf{v} - \frac{\mathbf{u}+\mathbf{w}}{2}\right)\right], \tag{5.61}$$

allowing to compute the changes in depth by minimizing the depth residual ρ_Z.

nonrigid-based energy minimization

The minimization of (5.53) is done per-pixel by solving:

$$\min_{\mathbf{w}}\ \Psi\left(\left|\mathbf{Rig}_{\mathbf{w}}^N - \mathbf{w}\right|^2\right) + \frac{\eta}{2}|\mathbf{w}-\mathbf{v}|^2, \tag{5.62}$$

where the parameter $\mathbf{x}$ is considered implicit. By setting the initial estimate as $\mathbf{w} = \mathbf{v}$, the scene flow can be iteratively computed via $\mathbf{w} \leftarrow \mathbf{w} + \Delta\mathbf{w}$, where the 3D motion increment, $\Delta\mathbf{w}$, is given by:

$$\Delta\mathbf{w} = \frac{2\Psi'\left(|\rho_{\mathrm{NR}}|^2\right)\rho_{\mathrm{NR}} + \eta(\mathbf{v}-\mathbf{w})}{2+\eta}, \tag{5.63}$$

with $\rho_{\mathrm{NR}} = \mathbf{Rig}_{\mathbf{w}}^N - \mathbf{w}$, the nonrigid residual, and Ψ' the robust norm derivative.

5.5 Coarse-to-fine estimation

Current RGBD sensors are able to provide more than 15 registered images per second, so that, inter frame 3D motions are expected in the range of few centimeters. However, the displacement on the image brought by the projection of the motion of a scene point becomes larger, the closer the 3D surface is from the camera, yielding quiet often to motions of more than one pixel of magnitude. Similarly, when the sensor itself moves rigidly relative to the scene, further scene points can induce large displacements in the image domain, even for a subtle camera motion. In such those cases and many others, images displacements are larger than one pixel between two frames and the scene flow energy is expected to be multi-modal, i.e. having multiple local minima. For this reason, the proposed minimization can be easily stuck in a local minimum. In order to find the global minimum, it is possible to apply multi-scale strategy (Anandan, 1989; Brox *et al.*, 2004): solving coarser and smoothed versions of the problem, each of which may have a unique minimum,

hopefully close to the global minimum of the original problem. In this strategy, every coarser version of the problem is formulated by downsampling the original images and its solution is used to initialize the solution of a finer version.

This multi-scale strategy is adopted in order to deal with larger motions. We construct a RGBD image pyramid with a downsampling factor of 2. A Gaussian anti-aliasing filter is applied to the intensity image at the original resolution and the intensity image pyramid is built using bicubic downsampling. For the depth image, a 5×5 median filter is used at the original resolution and the depth image pyramid is constructed by averaging pixels in non-overlapped neighborhoods of 2×2, where only pixels with a valid depth measure are used. Averaging on depth images is avoided to prevent the propagation of pixels having no valid depth measures. Having a pyramid with levels $l = \{0, 1, ..., L\}$, with 0 the original resolution, the computation is started at level L an the estimated twist field is directly propagated to the next lower level, as is shown in Figure 5.6, for a pyramid of 2 levels. The camera matrix is scaled at each level by the factor 2^l, compensating the changes in the projection of scene points due to downsampling. At each level M loops are performed, consisting of $M_{\mathbf{GN}}$ iterations of the Gauss-Newton procedure followed by $M_{\mathbf{TV}}$ iterations of the TV solver. The linking constant κ is styled at each scale. Algorithm 1 presents the general scheme of the scene flow estimation using the twist-motion representation. In order to reduce computational time, the domain of ξ is set to correspond to the domain of RGBD images at each level of the pyramid. For this reason, before moving to a finer level, a linear interpolation is applied on ξ to match the appropriate image domain.

The same multi-scale strategy can be applied if the 3D-motion representation is used. The scene flow $\mathbf{v}$ is filtering by component with a 5×5 median filter, before being propagated from one level to the finer one, robustly integratating flow estimates over large spatial neighborhoods (Sun *et al.*, 2010). Algorithm 2 presents the general scheme of the scene flow estimation using the 3D-motion representation.

5.6 Rigid plus nonrigid model

In many applications, the sensor itself moves relative to the observed scene and causes a dominant global motion in the overall motion field. In this situation, compensating for the motion of the camera can simplify the estimation and regularization of the scene flow. Figure 5.7 presents an example of how global rigid compensation modifies the motion estimation problem. Moreover, for some applications, such as SLAM and 3D reconstruction, the camera ego-motion is required. Particularly, for 3D reconstruction of deformable objects, the camera motion is needed to register partial 3D reconstructions, which are constructed by compensating the nonrigid motions. Therefore, we consider splitting the motion of the scene into a globally rigid component, $\xi_{\mathrm{R}} = (\tau_{\mathrm{R}}, \omega_{\mathrm{R}}) \in \mathbb{R}^6$, capturing the camera

Algorithm 1 Scene flow estimation using the twist-motion representation.

Super-scripted ^{l} denotes the pyramid level.

Input: two RGBD images $S_1 = \{I_1, Z_1\}$ and $S_2 = \{I_2, Z_2\}$
Output: scene flow $\mathbf{v}$ from S_1 to S_2

```
Construct RGBD pyramids S_1^l and S_2^l, l = 0, ..., L;
Initialize ξ = (ω, τ) = 0, χ = (ϖ, π) = 0, and l = L;
while l ≥ 0 do
  for w = 1 to M do
    for n = 1 to M_GN do
      Warp images I_2^l and Z_2^l using ξ; Eq. (4.3)
      Calculate residues ρ_I^l, ρ_g^l and ρ_Z^l;
      Update ξ; Eq. (5.38)
    end for
    for n = 1 to M_TV do
      Iterate once for solving ϖ; Eq. (5.42)
      Iterate once for solving π; Eq. (5.46)
    end for
  end for
end while
Compute the scene flow v from ξ; Eq. (4.2)
```

Algorithm 2 Scene flow estimation using the 3D-motion representation.

Super-scripted ^{l} denotes the pyramid level.

Input: two RGBD images $S_1 = \{I_1, Z_1\}$ and $S_2 = \{I_2, Z_2\}$
Output: scene flow $\mathbf{v}$ from S_1 to S_2

```
Construct RGBD pyramids S_1^l and S_2^l, l = 0, ..., L;
Initialize v = 0, w = 0, u = 0, and l = L;
while l ≥ 0 do
  for w = 1 to M do
    for n = 1 to M_GN do
      Warp images I_2^l and Z_2^l using ξ; Eq. (4.13)
      Calculate residues ρ_I^l, ρ_g^l and ρ_Z^l;
      Update v; Eq. (5.56)
    end for
    for n = 1 to M_TV do
      Iterate once for solving u; Eq. (5.42)
    end for
    Encourage local rigidity on w; Eq. (5.63)
  end for
  Median filter on v;
end while
```

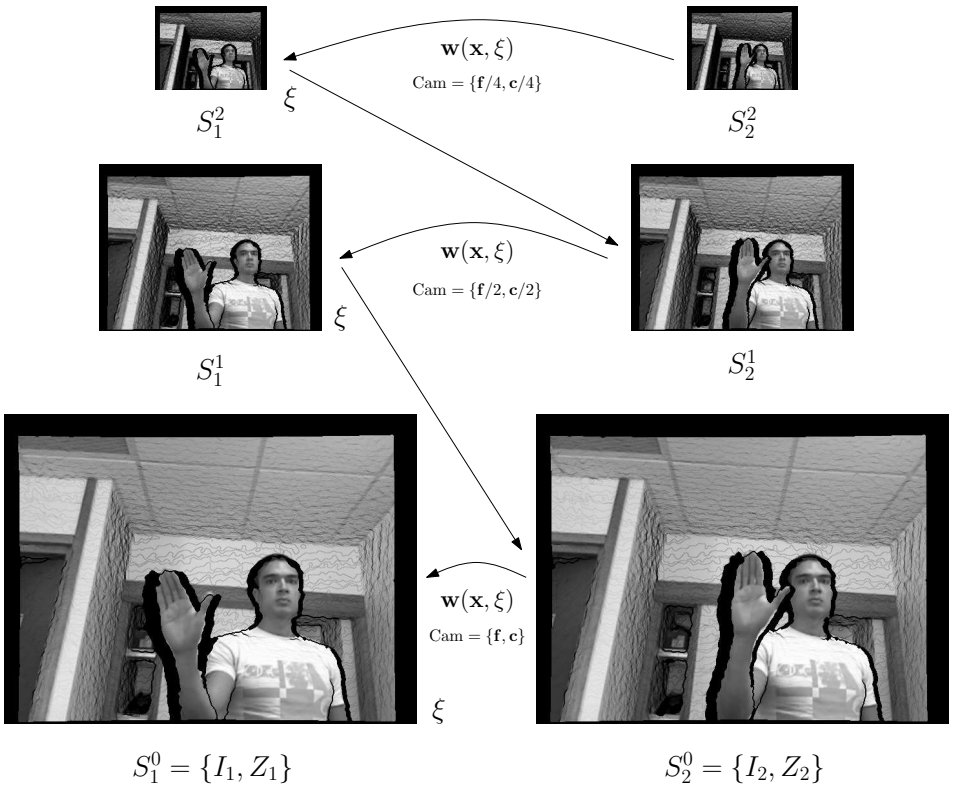


Fig. 5.6: *RGBD image pyramid of 2 levels. Intensity and depth pyramids are built with a downsampling factor of 2. Camera parameters* $\mathbf{f} = \{f_x, c_x\}$ *and* $\mathbf{c} = (c_x, x_y)$ *are scaled at each level* l*, with* $l = \{0, 1, 2\}$*, by the factor* 2^l*. The field of rigid motion* ξ *is directly propagate from one level to the next, starting from the coarsest level, given by images* $\{S_1^2, S_2^2\}$*, to the finest, images at the original resolution* $\{S_1^0, S_2^0\}$*.*

motion relative to the dominant object/background, plus a nonrigid residual field, $\xi = (\tau, \omega)$. We assume that a large part of the scene follows the same rigid motion. Accordingly, the scene flow can be recovered from the composition of twist fields given by $\chi = \log(e^{\hat{\xi}} + e^{\hat{\xi}_{\mathrm{R}}} - \mathbf{I}_{4\times4})$, and the estimation problem is formulated as:

$$\min_{\chi} \; E_{\mathbf{Rig}}(\chi) + E_{\mathbf{Res}}(\chi)\,, \tag{5.64}$$

with $E_{\mathbf{Rig}}(\chi)$ and $E_{\mathbf{Res}}(\chi)$ the rigid and nonrigid energies, respectively. It is worth noting that the separation of the camera motion is *in addition* to the framework presented above, i.e., the nonrigid part can still deal with local motion.

5.6.1 Rigid energy

The estimation of a global rigid motion can be seen as a particular case of the framework given by (5.2), where every twist of the motion field follows the same

Fig. 5.7: *Rigid motion compensation. From left to right, top to bottom: (a) first image, (b) second image, (c) difference between input images, (d) second image warped to compensate rigid motion, (e) image blended using first image and the compensated, and (f) difference between first image and the compensated. Once the rigid motion has been compensated the motion estimation becomes the computation of the nonrigid residual of the motion.*

rigid motion and so that, no regularization is needed. The camera motion, or the dominant rigid motion, can be estimated using the data term (5.14), by considering every pixel (or a subset of $\mathbf{\Omega}$) to solve for a unique twist ξ_{R} in a robust least-squares estimation. Accordingly, the rigid component of the energy is defined as:

$$E_{\mathbf{Rig}}(\chi) = \sum_{\mathbf{x}} w_I(\mathbf{x})\Psi\left(\rho_I^2\left(\mathbf{x},\chi\right) + \gamma\rho_g^2\left(\mathbf{x},\chi\right)\right) + w_Z(\mathbf{x})\Psi\left(\rho_Z^2\left(\mathbf{x},\chi\right)\right), \tag{5.65}$$

where $w_I(\mathbf{x})$ and $w_Z(\mathbf{x})$ weigh the per-pixel contribution of intensity- and depth-based terms. Setting $w_I(\mathbf{x}) = 1$ and $w_Z(\mathbf{x}) = \lambda$ yields the same formulation given by (5.12), but in general they can be defined as per-pixel functions of the confidence in intensity and depth data. Also it is possible to use only intensity or depth constraints. For example, setting $w_I(\mathbf{x}) = 0$ and $w_Z(\mathbf{x}) = 1$ corresponds to estimate a rigid transformation consistent only with the observed 3D surface.

5.6.2 Nonrigid energy

The residual nonrigid motion can be computed following the framework (5.2) for scene flow estimation. Accordingly, the nonrigid energy is given by:

$$E_{\mathbf{Res}}(\chi) = E_D(\chi) + \alpha E_S(\chi). \tag{5.66}$$

5.6.3 Rigid plus nonrigid estimation

The estimation of rigid and nonrigid components of the motion is a challenge since both are superimposed in the overall motion field affecting its separately estimation. If the rigid body motion is extremely dominant, covering by far most of the observed scene, it can be estimated independently as is done in the example of Figure 5.7. However, if the nonrigid motion is dominant, the independent estimation of the rigid motion is strongly biased by the nonrigid component. Similarly, the rigid component, especially when is brought by the camera motion, can produce large displacements on the image making harder the separately estimation of the nonrigid component. In order to reliably estimate the rigid and nonrigid components of the scene flow, we alternately estimate both components into the coarse-to-fine procedure. Accordingly, and knowing that $\chi = \log(e^{\hat{\xi}} + e^{\hat{\xi}_{\mathrm{R}}} - \mathbf{I}_{4\times4})$, energy (5.64) is minimized by the following iterative, alternating estimation of ξ_{R} and ξ:

a. Given a fixed ξ, solve for the *twist motion* ξ_{R} that minimizes $E_{\mathbf{Rig}}(\chi)$.

b. Given a fixed ξ_{R}, solve for the *twist field* ξ that minimizes $E_{\mathbf{Res}}(\chi)$.

Step **a**. is equivalent to minimize $E_{\mathbf{Rig}}(\xi_{\mathrm{R}})$ as is done in Section 5.4.1, but adding to the warp of every scene point $\mathbf{x}$, the 3D motion offset given by $(e^{\hat{\xi}} - \mathbf{I}_{4\times4})\tilde{\mathbf{X}}$, which compensates the nonrigid component. This minimization is done by iteratively applying (5.38), with a zero auxiliary flow, i.e., cropping the last term of equations (5.39) and (5.40). Accordingly, the increment $\Delta\xi_{\mathrm{R}}$ is computed as follows:

$$\Delta\xi_{\mathrm{R}} = -\mathbf{H}_{\mathrm{R}}^{-1}\mathbf{g}_{\mathrm{R}}, \tag{5.67}$$

where *gradient vector* $\mathbf{g}_{\mathrm{R}}$ is given by:

$$\mathbf{g}_{\mathrm{R}} = \sum_{\mathbf{x}\in\boldsymbol{\Omega}} \Psi'_{I,G}\left[(\mathbf{I}_{\mathbf{x}}\mathbf{J})^T\rho_I + \gamma(\mathbf{I}^g_{\mathbf{x}}\mathbf{J})^T\rho_g\right] + \lambda\,\Psi'_Z\,(\mathbf{Z}_{\mathbf{x}}\mathbf{J} - \mathbf{K})^T\,\rho_Z,$$

and the approximation of the Hessian matrix is given by:

$$\mathbf{H}_{\mathrm{R}} = \sum_{\mathbf{x}\in\boldsymbol{\Omega}} \Psi'_{I,G}\left[(\mathbf{I}_{\mathbf{x}}\mathbf{J})^T\,(\mathbf{I}_{\mathbf{x}}\mathbf{J}) + \gamma\,(\mathbf{I}^g_{\mathbf{x}}\mathbf{J})^T\,(\mathbf{I}^g_{\mathbf{x}}\mathbf{J})\right] + \lambda\,\Psi'_Z\,(\mathbf{Z}_{\mathbf{x}}\mathbf{J} - \mathbf{K})^T\,(\mathbf{Z}_{\mathbf{x}}\mathbf{J} - \mathbf{K}).$$

On the other hand, step **b**. is equivalent to minimize $E_{\mathbf{Res}}(\xi)$ as in Section 5.4, by iterating steps **i** and **ii**. The 3D motion offset, given by $(e^{\hat{\xi}_{\mathrm{R}}} - \mathbf{I}_{4\times4})\tilde{\mathbf{X}}$, should be added into the warping function of point $\mathbf{x}$, to compensate for the rigid component.

Algorithm 3 presents the general scheme of the scene flow estimation by modeling motion as a global rigid component plus a nonrigid residual.

Algorithm 3 Rigid plus nonrigid scene flow estimation.

Input: two RGBD images $S_1 = \{I_1, Z_1\}$ and $S_2 = \{I_2, Z_2\}$
Output: nonrigid scene flow $\mathbf{v}$ and global rigid motion ξ_{R}, from S_1 to S_2

```
Construct RGBD pyramids S_1^l and S_2^l, l = 0, ..., L;
Initialize ξ_R = 0, ξ = 0, and l = L;
while l ≥ 0 do
  for k = 1 to K do
    for w = 1 to M do
      for n = 1 to M_GN do
        Warp images I_2^l and Z_2^l using χ = log(e^ξ̂ + e^ξ̂R − I_4×4); Eq. (4.3)
        Calculate residues ρ_I^l, ρ_g^l and ρ_Z^l;
        Update ξ; Eq. (5.38)
      end for
      for n = 1 to M_TV do
        Iterate once for solving ϖ; Eq. (5.42)
        Iterate once for solving π; Eq. (5.46)
      end for
    end for
    for n = 1 to M_GN do
      Warp images I_2^l and Z_2^l using χ = log(e^ξ̂ + e^ξ̂R − I_4×4); Eq. (4.3)
      Calculate residues ρ_I^l, ρ_g^l and ρ_Z^l;
      Update ξ_R; Eq. (5.67)
    end for
  end for
end while
Compute the scene flow v from ξ; Eq. (4.2)
```

6 Experiments

6.1 Middlebury datasets

The Middlebury stereo dataset (Scharstein and Szeliski, 2003) is commonly used as benchmark to compare scene flow methods (Huguet and Devernay, 2007; Basha *et al.*, 2010; Hadfield and Bowden, 2011, 2014; Quiroga *et al.*, 2012, 2013, 2014a; Hornacek *et al.*, 2014). Using images of one of these datasets is equivalent to a fixed camera observing an object moving in X direction. As in previous methods, we take images 2 and 6 as the first and second RGBD images, respectively. Gray images are obtained from the RGB images, while the ground truth disparity map of each image is used as depth channel, see Figure 6.1. These datasets present a simple 3D motion field, where the ground truth for the scene corresponds to the camera motion along the X axis, which coincides with the baseline of the stereo setup. For this reason, Y- and Z-components of the scene flow are zero while the X-component is constant. On the other hand, the optical flow is not constant and is given by the disparity map of the first image, as is shown in Figure 6.2. For the experiments we consider three Middlebury datasets: *Teddy*, *Cones* and *Venus*. Particularly, Teddy and Cones datasets present a large displacement optical flow, whose magnitude in pixels is in the range $(12.5, 52.75)$ and $(5.5, 55)$, respectively.

6.1.1 Error measures

Error metrics for optical flow

In order to compare with previous methods, the scene flow error is firstly measured in the image domain using the *root mean squared error* (RMSE) and the *average angle error* (AAE) of the optical flow. Let (u, v) and $(u_{\mathrm{GT}}, v_{\mathrm{GT}})$ denote the image flow vector and the ground-truth flow vector, respectively. The *angular error* (AE)

Fig. 6.1: *Teddy stereo dataset. From left to right: (a) first and (b) second RGBD image pairs. Every depth image is computed from the disparity map of the corresponding view.*

is the angle in 3D space between $(u, v, 1)$ and $(u_{\mathrm{GT}}, v_{\mathrm{GT}}, 1)$, which is computed as:

$$\mathrm{AE} = \cos^{-1}\left(\frac{u \cdot u_{\mathrm{GT}} + v \cdot v_{\mathrm{GT}} + 1}{\sqrt{u^2 + v^2 + 1}\sqrt{u_{\mathrm{GT}}^2 + v_{\mathrm{GT}}^2 + 1}}\right).$$

Note that the AE corresponds to an angle in the range $[0, \pi]$. Accordingly, the AEE is defined by:

$$\mathrm{AAE} = \frac{1}{|\boldsymbol{\Omega}_{\mathrm{v}}|} \sum_{\mathbf{x} \in \boldsymbol{\Omega}_{\mathrm{v}}} \mathrm{AE}(\mathbf{x}), \tag{6.1}$$

with $|\boldsymbol{\Omega}_{\mathrm{v}}|$ the cardinality of set $\boldsymbol{\Omega}_{\mathrm{v}}$, which is defined as the subset of $\boldsymbol{\Omega}$ containing the non-occluded pixels having a valid depth measure.

The *endpoint error* (EE) is defined as the Euclidean distance on the image between (u, v) and $(u_{\mathrm{GT}}, v_{\mathrm{GT}})$, and is given by:

$$\mathrm{EE} = \sqrt{(u - u_{\mathrm{GT}})^2 + (v - v_{\mathrm{GT}})^2}.$$

Fig. 6.2: *Optical flow for the Teddy dataset. From left to right: (a) Blend of first and second brightness images, (b) ground truth of the optical flow, and (c) optical flow color coding. Observe that every pixel performs a purely horizontal motion, whose magnitude is a function of its depth.*

Using the EE, the RMSE of the optical flow is defined by:

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\mathbf{\Omega}_\mathrm{v}|} \sum_{\mathbf{x} \in \mathbf{\Omega}_\mathrm{v}} \mathrm{EE}(\mathbf{x})^2}. \tag{6.2}$$

Error metric for changes in depth

In order to measure how consistent is Z-component of the scene flow with the changes in depth, we use the *root mean squared error of the Z-motion* (RMSE_Z). This measure has been used to compare scene flow methods (Hadfield and Bowden, 2011, 2014; Hornacek *et al.*, 2014). Being $\mathbf{u}_\mathrm{GT}$ the ground-truth flow vector and v_Z the Z-component of the 3D motion field $\mathbf{v}$, and with $\{Z_1, Z_2\}$, the first and second depth images, the RMSE-Z is defined by:

$$\mathrm{RMSE}_Z = \sqrt{\frac{1}{|\mathbf{\Omega}_\mathrm{v}|} \sum_{\mathbf{x} \in \mathbf{\Omega}_\mathrm{v}} \left[(Z_2(\mathbf{x} + \mathbf{u}_\mathrm{GT}) - Z_1(\mathbf{x})) - v_Z(\mathbf{x}) \right]^2}, \tag{6.3}$$

Error metrics for scene flow

The computation of 2D motion errors does not necessarily reflect the errors in the scene since even small 2D displacements can produce large 3D motions. Moreover, it is preferable to compute 3D motions errors directly rather than measure independently errors of the optical flow and changes in depth, which can partially hide 3D motion errors. For instance, in Middlebury datasets the Z-component of the motion is zero, so that any scene flow estimation having no changes in depth will achieve a perfect score in RMSE_Z. In order to measure the overall scene flow error we define the *average normalized error* of the scene flow ($\mathrm{ANE}_\mathbf{V}$). The point-wise scene flow error is defined as the Euclidean distance between the

estimated 3D motion vector, $\mathbf{v} = (v_X, v_Y, v_Z)$, and the ground-truth 3D motion, $\mathbf{v}^{\text{GT}} = (v_X^{\text{GT}}, v_Y^{\text{GT}}, v_Z^{\text{GT}})$. Subsequently, every error measure is normalized by the magnitude of the ground-truth scene flow, which is assumed to be non-zero, to present the error as a percentage of the motion magnitude. Accordingly, the $\text{ANE}_{\mathbf{V}}$ is defined by:

$$\text{ANE}_{\mathbf{V}} = \frac{1}{|\mathbf{\Omega}_{\text{v}}|} \sum_{\mathbf{x} \in \mathbf{\Omega}_{\text{v}}} \frac{\sqrt{(v_X - v_X^{\text{GT}})^2 + (v_Y - v_Y^{\text{GT}})^2 + (v_Z - v_Z^{\text{GT}})^2}}{|\mathbf{v}^{\text{GT}}|}. \tag{6.4}$$

Particularly, for the Middlebury datasets the magnitude of the scene flow is constant and is given by the baseline of the stereo setup. Additional to the $\text{ANE}_{\mathbf{V}}$, we compute the statistic Rp% to show the percentage of pixels having a normalized scene flow error smaller or equal to p% of the ground truth magnitude.

6.1.2 Baseline methods

We consider two baseline methods that follows the proposed framework presented in Chapter 5: *6TwistFlow* and *3DmotionFlow*. We named 6TwistFlow the method that uses the twist-motion representation to minimize the scene flow energy (5.2). The algorithm using the 3D-motion representation to minimize energy (5.49), is called 3DmotionFlow. The main parameters are listed in Table 6.1.

For both baseline methods we set the following parameters the same: $L = 5$, $\alpha = N^2$ $\beta = 0$, $\gamma = 0$, $s_{\text{tr}}^2 = \max$ (off), $M_{\mathbf{TV}} = 50$, $M_{\mathbf{GN}} = 5$. Now, for the 6TwistFlow method we independently set: $\lambda = 0.1$, $N = 5$, $M = 5$, and $\kappa(l) = N^{-2}10^{-3}10^{l}$. Finally, for the 3DmotionFlow we set: $\lambda = 1$, $N = 11$, $M(l) = \{1, 2, 3, 4, 5, 5\}$, $\zeta = 0$ (the local-rigid prior set off in E_S), $N_\zeta = 5$, and $\kappa(l) = N^{-2} \times \{2^{-2}, 10^{-1}, 10^{0}, 10^{2}, 10^{4}, 10^{6}\}$. For both methods, the weighting function in the data term (5.12) follows a uniform distribution.

6.1.3 Comparison with other methods

We consider three groups of methods in the comparison: *i*) optical flow methods, *ii*) stereo-based methods and *iii*) RGBD-based methods.

Optical flow methods

Optical flow methods use only the RGB information of first and second images to estimate the 2D motion. The scene flow is computed by inferring the 3D motion with the provided depth data, as is shown in Table 4.1 for the 2D-motion representation. We consider the *large displacement optical flow* algorithm by Brox and Malik (2011) (*OpticalFlow1*) and the *motion detail preserving* algorithm by Xu *et al.* (2012)

L	maximum level of the RGBD pyramid
α	smoothing parameter
β	decreasing parameter of the adaptive TV
γ	gradient constraint parameter
λ	depth constraint parameter
κ	linking parameter
s^2_{tr}	bound of the robust norm
N	size of the local neighborhood, $N \times N$ centering window
M	number of alternations in each level of the pyramid
$M_{\mathbf{TV}}$	iterations of the TV solver
$M_{\mathbf{GN}}$	iterations of the Gauss-Newton algorithm
ζ	local-rigidity parameter (3DmotionFlow)
N_ζ	size of the local neighborhood for the rigid fitting (3DmotionFlow)

Table 6.1: *Parameters of baseline scene flow methods.*

(*OpticalFlow2*). Comparison with optical flow methods allow to asses the relative contribution of using depth data in addition.

Stereo-based methods

Stereo-based methods do not assume RGBD images and simultaneously estimate the optical flow and disparity maps by considering RGB images 2, 4, 6 and 8 of each dataset. Images 2 and 4 form the first stereo pair, and images 6 and 8 the second one. We consider the *variational method from stereo* by Huguet and Devernay (2007) (*StereoFlow1*), which was the first to use these datasets for scene flow, and the *multi-view variational* algorithm by Basha *et al.* (2010) (*StereoFlow2*).

Scene flow methods

RGBD-based methods, as those proposed in this thesis, use simultaneously RGB and depth images for the estimation. We consider two scene flow methods: the *particle filter based* algorithm by Hadfield and Bowden (2014) (*SceneFlow1*) and the *sphere flow* algorithm by Hornacek *et al.* (2014) (*SceneFlow2*).

Results

Table 6.2 presents the results on the Middlebury datasets for the eight algorithms considered. The proposed algorithm 6TwistFlow achieves the best results for RMSE and RMSE_Z, confirming the benefit brought by the twist representation.

	Teddy			Cones			Venus		
	RMSE	AAE	$RMSE_Z$	RMSE	AAE	$RMSE_Z$	RMSE	AAE	$RMSE_Z$
OpticalFlow1	2.83	0.39	1.75	3.20	0.39	0.47	0.72	1.28	0.14
OpticalFlow2	1.66	**0.21**	1.15	1.70	**0.28**	0.50	0.3	1.43	0.22
StereoFlow1	1.10	0.69	n/a	1.25	**0.51**	n/a	0.31	0.98	n/a
StereoFlow2	0.58	0.39	n/a	0.57	1.01	n/a	0.16	1.58	n/a
SceneFlow1	1.24	1.01	0.06	0.83	0.83	0.03	0.36	1.03	0.02
SceneFlow2	**0.54**	0.52	**0.02**	**0.35**	**0.15**	**0.01**	0.26	**0.53**	0.02
6TwistFlow	**0.35**	**0.35**	**0.02**	**0.40**	**0.34**	**0.02**	**0.15**	**0.84**	**0.01**
3DmotionFlow	0.71	0.67	**0.04**	0.51	0.31	0.04	**0.2**	1.02	**0.00**

Table 6.2: *Comparison with other methods using Middlebury datasets. First and second best performances are highlighted in red and orange, respectively.*

Observe that in most cases this method outperforms SceneFlow2, which also use a field of rigid motion, but in a discrete optimization scheme, which could benefit from the propagation of the correct rigid motion. AEE results of both methods are very similar, differing only in about 1/4 of degree. On the other hand, the alternative approach 3DmotionFlow outperforms most of the considered methods, except SceneFlow2. Particularly, it is interesting to remark that 3DmotionFlow outperforms SceneFlow2, which directly represent the scene in 3D and solve for the scene flow testing and propagating 3D motion hypothesis. Although the 3D motion of every scene point is the same and few correct hypothesis could favor the final solution through the particle filtering, the incremental 3D motion estimation done by 3DmotionFlow performs better. Moreover, the 2D parametrization of the scene, used both by the 6TwistFlow and by the 3DmotionFlow algorithms, allows to accurately solve for the 3D motion field. It is important to remark that SceneFlow1 and SceneFlow2 use a 3D representation of the scene, as a 3D point cloud and a set of 3D patches, respectively. Therefore, there is no any advantage of using a 3D representation for the motion estimation on these datasets.

Regarding stereo-based methods, the measured errors are higher in most cases since the depth information is estimated at the same time. However, thanks to the two additional views these methods benefits from smaller occluded regions, and the constraints brought by the stereo setup favor the estimation of the purely horizontal 2D motion. Finally, optical flow methods are the worst performer, which is expected since they only use the brightness of two views. Particularly, Teddy and Cones datasets are challenging motion estimation problems because the observed optical flow presents a wide range, around 40 and 50 pixels, respectively. Unlike their RMSE is between the highest errors, for both methods, the AEE is very low. This accurate angular estimation is explained by the strong TV regularization done directly on the optical flow and because all pixels move in the same direction. Finally, it is observed from Table 6.2 that the critical measure for optical flow

methods is the estimation of changes in depth, although there is no Z-motion in these datasets. Due to the lack of depth information in the motion estimation, even small errors on the image motion yield to large error in the 3D motion inferred using the provided depth data. Accordingly, the jointly use of brightness and depth data allows to a more accurate estimation of motion on the image domain, and simultaneously, changes in depth.

6.1.4 Analysis of the scene flow framework

The proposed scene flow framework allows the implementation of different approaches through the setting of parameters. For example, local and piecewise rigidity can be adjusted by varying the size of the local neighborhood, $N(\mathbf{x})$, and the weight of the smoothness term, α, respectively. Similarly, it is possible to configure the balance between several terms, such as brightness and gradient constraints, brightness- and depth-based constraints, and data en smoothness energies. We analyze below how the configuration of some of these parameters affects the performance of the scene flow framework on Teddy and Cones datasets. Additional to RMSE, AAE and RMSE_Z, we compute $\text{ANE}_{\mathbf{V}}$ and R5% to directly compare 3D motion errors.

Local rigidity

The scene flow is assumed to be semi-rigid, i.e., local and piecewise rigid. While 6TwistFlow directly encourages local rigidity in data term, 3DmotionFlow allows to incorporate the local rigidity in both terms: in the data term it is done by encouraging a constant 3D translation, while in the smoothness term a local rigid motion is fitted and the residual motion minimized. Using baseline methods, we perform an experiment varying the size of $N(\mathbf{x})$, thereby controlling the amount of local rigidity assumed in the data term. We start from $N(\mathbf{x}) = \{\mathbf{x}\}$, i.e., no local rigidity, to $N(\mathbf{x})$ being a 11×11 window, as is presented in Table 6.3. In the 3D-motion case, the bigger the neighborhood the more accurate the results, showing that a strong local rigidity encouragement favors the scene flow estimation. On the other hand, the best performance is obtained for middle size windows, 5×5 and 7×7. Although RMSE subtly decreases as the size of the window increases, $\text{ANE}_{\mathbf{v}}$ and AAE present an inflection point around 5×5, as is shown in Figure 6.3. This is a meaningful result for 6TwistFlow, showing that a bigger local neighborhood does not imply a better estimation of the 3D motion, even for this constant scene flow. Moreover, unlike 3DmotionFlow, accurate results for 6TwistFlow start from with a small window, 3×3, therefore it can be inferred that the amount of local rigidity assumption has less significance in the final results.

Particularly for 3DmotionFlow, local rigidity can be encouraged in data and smoothness terms. In order to asses the benefit of each strategy, we consider

		Teddy				Cones			
	$N(\mathbf{x})$	RMSE	AAE	$ANE_{\mathbf{v}}$	P5%	RMSE	AAE	$ANE_{\mathbf{v}}$	P5%
3DmotionFlow	1×1	9.81	5.21	19.1	43.17	13.1	8.70	29.1	29.4
	3×3	1.83	1.58	4.16	78.26	1.66	1.11	3.22	87.1
	5×5	1.30	1.18	3.28	82.65	0.99	0.57	2.15	90.1
	7×7	1.02	0.918	2.71	85.74	0.73	0.51	1.91	94.0
	9×9	0.81	0.76	2.30	87.92	0.71	0.44	1.73	95.2
	11×11	**0.71**	**0.67**	**2.01**	**89.69**	**0.51**	**0.34**	**1.37**	**98.1**
6TwistFlow	1×1	1.26	0.96	5.14	52.1	1.60	1.02	7.60	34.6
	3×3	0.44	0.46	2.37	97.9	0.40	**0.30**	1.32	**100.**
	5×5	0.43	0.33	**1.57**	**100.**	0.41	0.33	**1.17**	**100.**
	7×7	**0.35**	0.35	1.63	**100.**	0.40	0.34	1.22	**100.**
	9×9	0.38	0.32	1.80	**100.**	**0.39**	0.32	1.39	**100.**
	11×11	**0.35**	**0.31**	1.74	**100.**	0.40	0.31	1.8	**100.**

Table 6.3: *Variation of the local rigidity in the data term. Top results in red and blue for 3DmotionFlow and 6TwistFlow, respectively. Baseline methods are highlighted in green.*

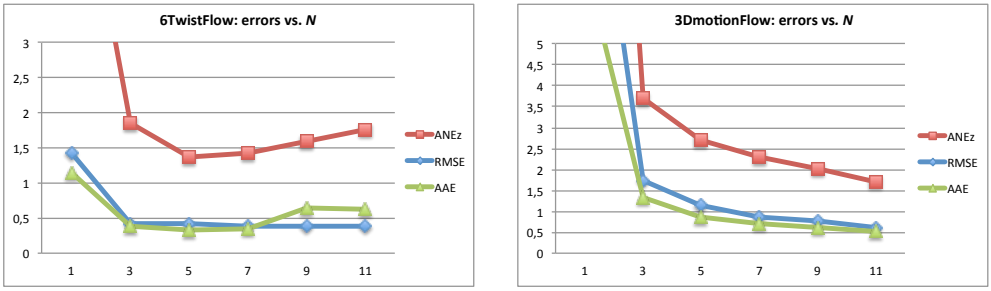


Fig. 6.3: *Averages of Teddy and Cones datasets for the motion errors RMSE, AAE and ANE_Z, as functions of the size of the local neighborhood.*

two variations of the baseline method: *3DmotionE$_S$* and *3DmotionALL*. The first, 3DmotionE$_S$, removes the local encouragement in the data term by setting $N(\mathbf{x}) = \mathbf{x}$, i.e., one pixel neighborhood, and includes the local rigid prior in the smoothness term. We modify the linking parameter to improve the accuracy by setting $\kappa(l) = \{10^{-2}, 10^{-1}, 10^{0}, 10^{0}, 10^{1}, 10^{1}\}$, giving more weight to the regularization in order to improve the stability in the minimization of the data term. The second approach, 3DmotionALL, applies the local-rigid prior both in data and smoothness term, which is equivalent to add the local-rigid prior in the regularization of the baseline method. In both cases, the local fitting is done on a 11×11 neighborhood, i.e., $N_\zeta = 11$, and the local-rigid parameter is set as $\zeta = 3$. The use of the local-rigid prior in addition to the local estimation in the data term, yields to a more accurate estimation of the motion, as is presented in Table 6.4. On the other hand, the encouragement alone of the local motion seems to be more effective in the data

term than in the regularization procedure.

	Teddy				Cones			
	RMSE	AAE	$ANE_{\mathbf{v}}$	P5%	RMSE	AAE	$ANE_{\mathbf{v}}$	P5%
3DmotionFlow	0.71	0.67	2.01	89.7	0.51	0.34	1.37	98.1
3DMotionE_S	1.41	1.27	3.80	78.2	1.50	1.49	4.26	77.7
3DMotionALL	**0.68**	**0.64**	**1.98**	**89.9**	**0.48**	**0.32**	**1.30**	**98.2**

Table 6.4: *Local rigidity in the smoothness term. Top results are shown in red. The best results are achieved when the local-rigid prior in the smoothness term is applied in addition to the local rigidity assumption in the data term.*

Piecewise rigidity

Piecewise rigidity is encouraged in the smoothness term by TV regularization. We investigate the influence of this assumption in the performance of 6TwistFlow and 3DmotionFlow, by varying the number of iterations of the TV solver, $M_{\mathbf{TV}}$. As is shown in Table 6.5, a stronger piecewise rigidity yields to a better results since the global rigid motion of the dataset. Also observe that for $M_{\mathbf{TV}} = 0$, there is no TV regularization favoring piecewise rigidity but there exists a Tikhonov regularization on the energies from the linking term controlled by κ. This kind of regularization favors small norm solutions. In this case, unlike 6TwistFlow that provides totally wrong estimations, the performance of 3DmotionFlow is still acceptable. This is explained by the median filtering on $3DmotionFlow$, which is directly related with l^1 minimization (Sun *et al.*, 2010) and therefore favor piecewise smooth solutions.

Motion representation

This experiment compares the different motion representation. Alternatively to 6TwistFlow and 3DmotionFlow, it is also possible to use a simpler motion representation considering a orthographic camera, as in most optical flow methods. As is explained in Section 4.3.4, in this case, the depth data is no directly used to model the image motion. The proposed framework can be straightforwardly used with the 2D-motion representation. We named *2DmotionFlow* the method based on energy (5.49), with the warping function given by (4.27). This baseline method is set with $L = 5$, $\alpha = N^2$, $\beta = 0$, $\gamma = 0$, $s^2_{\text{tr}} = \max$ (off) $\lambda = 0.1$, $N = 7$, $M = 5$, $M_{\mathbf{GN}} = 5$, $M_{\mathbf{TV}} = 100$, $\zeta = 0$, $N_\zeta = 5$ and $\kappa(l) = N^{-2} \times \{2, 10^1, 20, 10^2, 10^3, 10^3\}$.

Table 6.6 shows errors for the three representations, and includes *GroundTruth*, which is obtained by applying the ground truth 3D motion given by the baseline of the stereo setup. Note that for GroundTruth the RMSE is not zero due to the dataset disparities, which are used to compute the ground truth optical flow, have quarter-pixel accuracy. As it was concluded from previous experiments, the method based on the twist representation performs better that the method using the 3D

		Teddy				Cones			
	$M_{\mathbf{TV}}$	RMSE	AAE	$\text{ANE}_{\mathbf{v}}$	P5%	RMSE	AAE	$\text{ANE}_{\mathbf{v}}$	P5%
3DmotionFlow	0	1.42	1.12	3.13	85.4	0.84	0.58	2.09	92.8
	10	1.14	0.97	2.71	86.7	0.72	0.5	1.86	1.14
	50	0.71	0.67	2.01	89.7	0.51	0.34	1.37	98.1
	100	**0.60**	**0.58**	**1.78**	**91.8**	**0.43**	**0.28**	**1.14**	**98.6**
6TwistFlow	0	15.5	14.7	24.9	36.1	21.6	23.3	40.0	18.6
	10	2.04	0.59	5.03	68.6	1.25	0.62	3.87	77.8
	50	0.35	0.35	1.63	100.	0.40	0.34	1.22	100.
	100	**0.33**	**0.28**	**1.29**	**100.**	**0.33**	**0.25**	**0.79**	**100.**

Table 6.5: *Variation of the piecewise rigidity assumption. Top results in red and blue for 3DmotionFlow and 6TwistFlow, respectively. Baseline methods are highlighted in green.*

	Teddy				Cones			
	RMSE	AAE	$\text{ANE}_{\mathbf{v}}$	P5%	RMSE	AAE	$\text{ANE}_{\mathbf{v}}$	P5%
GroundTruth	**0.13**	**0.00**	**0.00**	**100.**	**0.24**	**0.00**	**0.00**	**100.**
6TwistMotion	0.35	0.35	1.63	100.	0.40	0.34	1.22	100.
3DmotionFlow	0.71	0.67	2.01	89.7	0.51	0.34	1.37	98.1
2DmotionFlow	3.44	0.89	10.4	56.6	3.04	0.74	10.7	68.2

Table 6.6: *Comparison of motion representations. Groundtruth results are highlighted in red. The twist-motion representation achieves the best performance.*

representation. Observe that the use of the 2D-motion representation does not bring any advantage over optical flow methods for the estimation of the image motion, see Table 6.2. For this reason, the use of depth data only as an additional brightness channel, wastes useful information for the constraint of the image motion. On the other, the estimation of changes in depth by 2DmotionFlow outperforms optical flow methods; disparity errors are 0.06 and 0.12, for Teddy and Cones, respectively. This is thanks to the use of a depth constancy assumption, which allows to regularize the estimation of the Z-component of the scene flow. Conversely, optical flow methods only infer the 3D motion field from the provided depth data. Finally, optical flow results are presented in Figures 6.4 and 6.5.

Brightness and depth balance

The use of a warping function allows to constrain the scene flow on the image and therefore to exploit both intensity and depth data. We experiment by varying the balancing term λ to modify the contribution of brightness and data terms in the scene flow energy. We consider $\lambda = \{0, 0.1, 1, \infty\}$. Notation $\lambda = \infty$ stands for the case where only depth constancy constraints are used in the data term. Table 6.7 presents the results. It can be noted that the simultaneous constraint of the motion in brightness and depth achieves the best performance for each representation.

Fig. 6.4: *Optical flow estimates for the Teddy stereo dataset. From left to right, from top to bottom: (a) Ground-truth, and (b) twist-motion, (c) 3D-motion and (d) 2D-motion representations.*

		Teddy				Cones			
	λ	RMSE	AAE	ANE_v	P5%	RMSE	AAE	ANE_v	P5%
2DmotionFlow	0	**3.41**	0.94	14.2	**57.0**	3.12	0.79	26.1	66.6
	0.1	3.44	**0.89**	**10.4**	56.6	**3.04**	**0.74**	**10.7**	**68.2**
	1	3.63	1.42	11.6	50.9	4.06	0.78	12.3	64.8
	∞	9.25	9.78	28.0	28.4	7.07	2.92	46.7	25.0
3DmotionFlow	0	1.12	0.91	13.2	28.2	1.19	0.94	15.7	22.1
	0.1	0.98	0.76	2.38	89.1	0.72	0.60	1.85	93.3
	1	**0.71**	**0.67**	**2.01**	**89.7**	**0.51**	**0.34**	**1.37**	**98.1**
	∞	3.26	4.33	8.47	67.7	1.18	0.82	3.13	86.3
6TwistFlow	0	0.38	0.36	1.66	**100.**	0.47	0.42	4.09	70.2
	0.1	**0.35**	**0.35**	**1.63**	**100.**	**0.40**	0.34	1.22	**100.**
	1	0.54	0.57	2.21	93.17	0.45	**0.25**	**1.18**	99.2
	∞	0.88	0.59	3.5	70.3	1.28	1.19	3.85	67.5

Table 6.7: *Variation of the balance between brightness and depth. Top results in orange, red and blue for 2DmotionFlow, 3DmotionFlow and 6TwistFlow, respectively. Baseline methods are highlighted in green. The simultaneous constraint in brightness and depth achieves the best results.*

Particularly, for the twist representation there is subtle difference between the baseline method and the constraint only on the brightness image. This can be

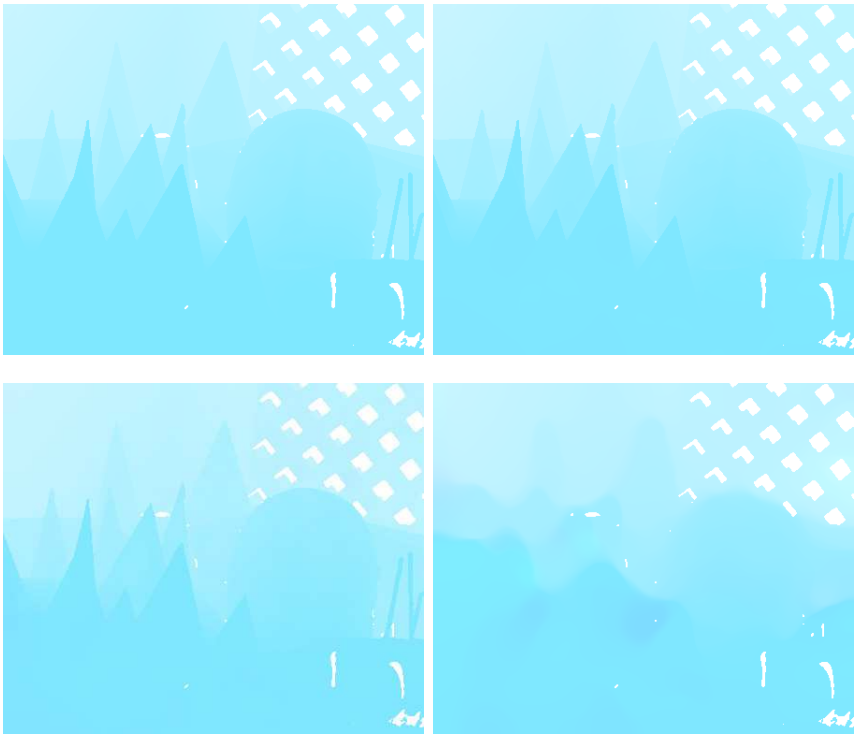


Fig. 6.5: *Optical flow estimates for the Cones stereo dataset. From left to right, from top to bottom: (a) Ground-truth, and (b) twist-motion, (c) 3D-motion and (d) 2D-motion representations.*

explained by the zero Z-component of the scene flow and by the ability of the twist representation to constrain this simple motion on the brightness image. However, the same difference becomes larger for the 3D-motion representation. On the other hand, for the 2D-motion representation the addition of depth constraints brings a very slight advantages respect of the optical flow case ($\lambda = 0$), even this latter performs better in some cases. The constraint of the motion only on depth data is the worst performer in each case, since the texture of depth images is by far less meaningful that such of brightness images. Finally, it is important to remark that the best balance varies from one motion representation to another, and therefore it is no possible to conclude about the optimal combination.

Processing time and resolution

We measured the runtime of non-optimized implementations of baseline methods 3DmotionFlow and 6TwistFlow, using the single-core 64-bit Intel Xeon E5520 @ 2.27GHz with 8 Gb of RAM. Table 6.8 summarizes the computational times of both approaches, and includes the runtimes reported of SceneFlow1 (Hadfield and Bowden, 2014), which also uses a single-core machine. Proposed methods 3DmotionFlow and 6TwistFlow requires less than 1/10 and 1/5 of runtime than

	Teddy	Cones
3DmotionFlow	42.2 s	44.3 s
6TwistMotion	80.5 s	79.4 s
SceneFlow1	418. s	493. s

Table 6.8: *Runtime comparison.*

SceneFlow1, respectively, and both achieve better estimations. Also observe that thanks to its simpler parametrization, 3DmotionFlow runs almost twice faster than 6TwistFlow. Comparison with SceneFlow2 was not possible because its runtime is not reported (Hornacek *et al.*, 2014), but authors claim it takes about 300 s. Computational speed is not our primary concern but there are several ways to speed up the scene flow methods. Particularly, we focus in a simple modification using our implementations in the same machine. For this purpose, we modify the resolution of the scene representation to get faster estimations, and at the same time, we measure the accuracy of the obtained estimate. First, we experiment with an early stop of the coarse-to-fine procedure as follows. Being at level $l = \text{PyrStop}$ of the RGBD pyramid, the estimation is stopped and the final motion field at the original resolution is obtained by linear interpolation. Table 6.9 presents the average results on Teddy and Cones datasets. Observe that with $\text{PyrStop} = 1$, the computation is stopped at the first octave (half of the initial scale) and runtimes are reduced by about 2 and 4, for 3DmotionFlow and 6TwistFlow, respectively, while the accuracy is almost equivalent to that obtained at the original resolution.

Finally, we consider only every mth pixel of the motion field for the Gauss-Newton procedure in the data term. The result of each operation is copied to the surrounding pixels. For example, setting $m(l) = \{4, 4, 2, 2, 1, 1\}$ as a function of the pyramid level l, for 3DmotionFlow, the runtime is reduced to 5.9 s, i.e., about 7 times faster, while the accuracy stands almost unchanged. Similarly, setting $m(l) = \{6, 6, 4, 4, 2, 2\}$ for 6TwistFlow, the runtime becomes 13.8 s, i.e., about 6 times faster, with nearly the same accuracy.

	3DmotionFlow					**6TwistFlow**				
PyrStop	RMSE	AAE	$\text{ANE}_\mathbf{v}$	P5%	Time(s)	RMSE	AAE	$\text{ANE}_\mathbf{v}$	P5%	Time
5	5.30	4.91	17.9	6.64	0.13	6.79	4.73	19.9	0.00	0.61
4	2.01	1.84	7.46	43.8	0.60	1.19	1.01	4.19	75.3	0.79
3	0.90	0.81	3.09	79.7	2.25	0.57	0.55	2.08	99.2	1.69
2	0.71	0.64	2.09	92.8	7.07	0.45	0.40	1.64	100.	5.30
1	0.62	0.52	1.84	93.0	19.3	0.40	0.36	1.47	100.	20.2
0	0.61	0.50	1.69	93.9	43.3	0.37	0.34	1.42	100.	79.9

Table 6.9: *Early stopping in the coarse-to-fine procedure. PyrStop stands for the level of the pyramid where the computation is stopped. The resulting motion field is linearly interpolated to compute the final motion field.*

6.2 Scene flow from RGBD images

We performed further experiments on more complex scenes using two RGBD sensors: the Microsoft Kinect for Xbox and the Asus Xtion Pro Live. We consider three different setups: *i*) a fixed or moving camera assuming a nonrigid motion, *ii*) a fixed or moving camera observing a rigid scene and *iii*) a moving camera capturing deformable objects. In each case, we show the input images, the optical flow, and one or more components of the scene flow. Also, depending of the experiment we present the warped images, the resulting brightness and depth residuals and the 3D reconstruction. The following example presents a scene flow estimation example.

A fixed camera is observing a non rigid scene, as is showed in Figure 6.6. The scene is segmented in the image domain using the observed depth measures of the first depth image. For this example, the estimation is done only for scene points within 50 cm and 200 cm from the RGBD sensor. The bottom images in Figure 6.6 give an idea of the segmentation. It is assumed that the 3D surfaces of interest are within this given range. No motion estimation is done for the remaining points. Nevertheless, all points of the second image, having a valid depth measure, are considered for the scene flow estimation. Results are presented as is shown Figure 6.7. The optical flow is visualized using the Middlebury color code (Scharstein and Szeliski, 2003). For the scene flow, we show each component using a cold-to-warm code, where green color is zero motion, and warmer and colder colors

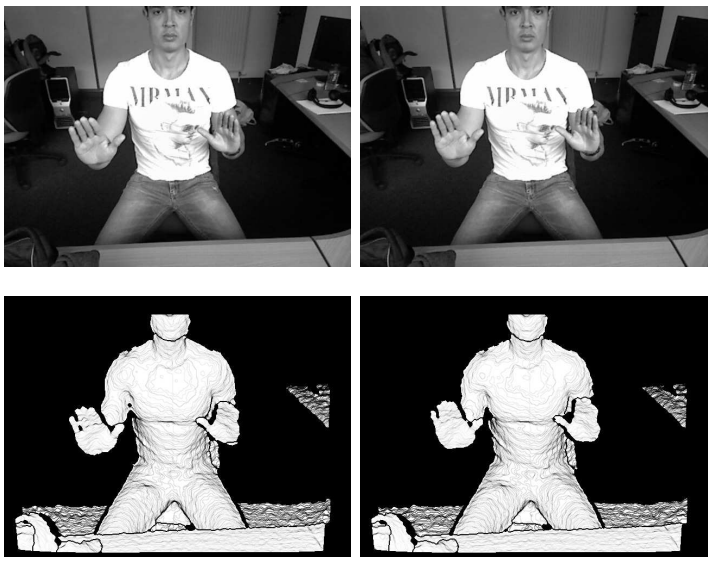


Fig. 6.6: *Inputs for scene flow estimation. From left to right: (a) brightness and (b) 3D surface of the first and second frames. The 3D surface is illustrated using directly the depth image, where a decreasing function of the magnitude of the gradient modulates the visualization, giving an idea of the 3D structure. Only the 3D surface within 50 cm and 200 cm from the sensor is shown.*

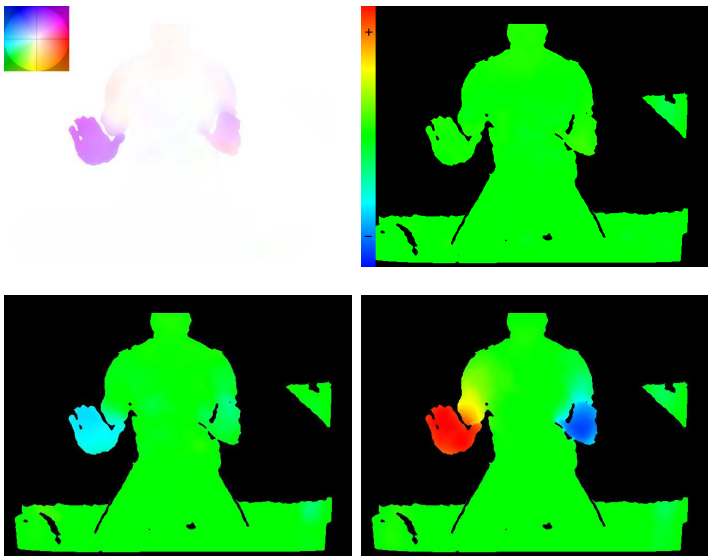


Fig. 6.7: *Scene flow results. From left to right, and from top to bottom: (a) optical flow, and (b) X-, (c) Y- and (d) Z-components of the scene flow, respectively.*

represent positive and negative velocities, respectively. Finally, Figure 6.8 shows the brightness and depth residuals, which correspond to the differences between the original and warped RGBD representations of the scene.

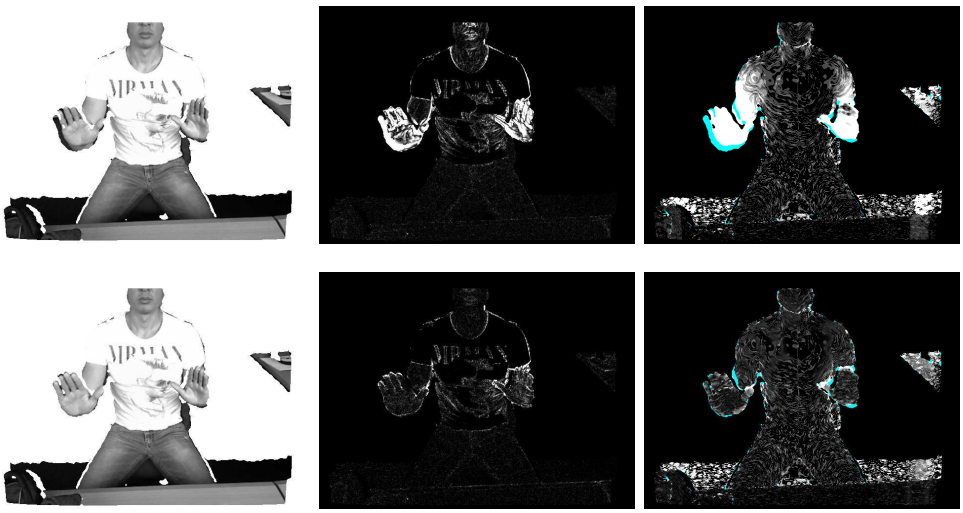


Fig. 6.8: *Brightness and depth residuals. From left to right: (a) warped image, and (b) brightness and (c) depth residuals, respectively. The top row shows results assuming a null scene flow. The bottom row presents the warped image and differences with the estimated scene flow. Brightness differences are scaled by 8 for visualization, and depth differences go from black to white, for 0 cm to 1 cm, and which for larger differences. Cyan color stands for an invalid depth difference.*

6.2.1 Nonrigid motion estimation

In this section we perform some nonrigid estimation experiments. Firstly, we consider a large Z-motion case presented by Hornacek *et al.* (2014), where it is possible to compare some of the current scene flow methods. Secondly, four further experiments are performed using the proposed approaches.

Large Z-motion experiment

Hornacek *et al.* (2014) propose to estimate in both directions, the motion represented by the pair of RGBD images shown in Figure 6.9. This is a challenging setup due to the large displacement in Z-direction, illumination changes, partial occlusions and the low resolution of the RGB images, 320×240 pixels. Results of four scene flow methods are presented in Figure 6.10, including our approach (Quiroga *et al.*, 2013), which corresponds to the baseline method 3DmotionFlow in the previous section. Observe that (Herbst *et al.*, 2013) and (Hadfield and Bowden, 2014) fail capturing the coarse motion, while (Hornacek *et al.*, 2014) and (Quiroga *et al.*, 2013) are able to give a rough idea of the observed motion. However, it is not possible to measure the accuracy using only the estimated optical flow. We replicated the experiments using the methods 6TwistFlow and 3DmotionALL, and also include the estimation using the large displacement method (LDOF) by Brox and Malik (2011), to have an idea of the performance of current optical flow methods. Results are presented in Figure 6.11, where it can be seen that all methods are able to capture the rough motion with some small differences.

Fig. 6.9: *Large Z-motion experiment proposed by Hornacek et al. (2014). From left to right: (a) color image, (b) 3D surface and (c) RGBD image. First and second images are presented in the top and bottom rows, respectively. The experiment estimates the motion in both directions.*

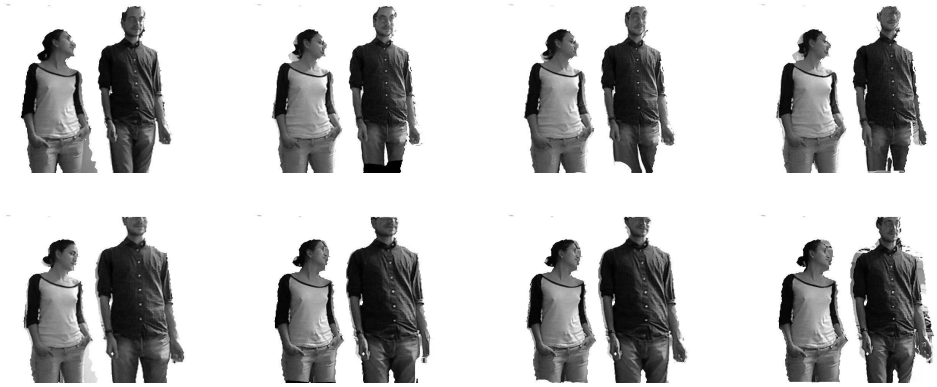
(a) Hornacek *et al.* (2014) (b) Hadfield and Bowden (2014) (c) Herbst *et al.* (2013) (d) Quiroga *et al.* (2013)

Fig. 6.10: *Optical flow results of the large Z-motion experiment presented by (Hornacek et al., 2014). The top images present the estimation from frame 1 to 2, while the bottom images for the contrary sense. Observe, that only (a) and (b) are able to estimate the motion in both cases.*

(a) 6TwistFlow (b) 3DmotionALL (c) LDOF

Fig. 6.11: *Optical flow results of the large Z-motion experiment. All methods are able to overcome the large change in depth and give a roughly estimation of the performed motion.*

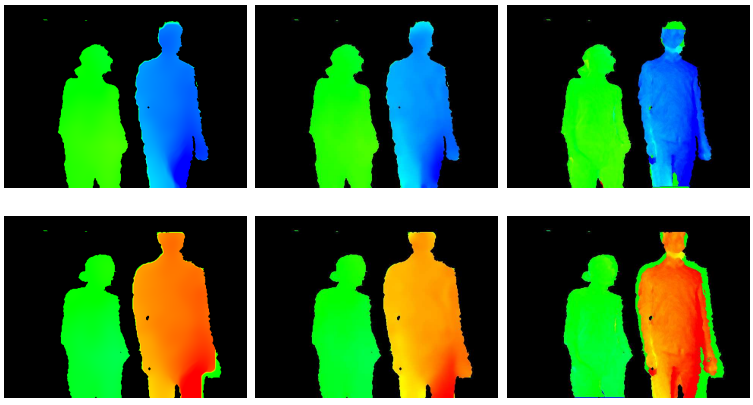
In order to have an idea of the accuracy, we warp second images using the estimated motion. The resulting warped images are shown in Figure 6.12. It can be seen that 6TwistFlow models the best the large Z-motion performed by the person on the right, especially visible in the warped texture of the face. However, there exist some problems estimating the motion of one of the hands in the second case, due to the low resolution of depth data and the splitting of the surface from frame 2 to 1. On the other hand, 3DmotionALL is able to model the motion of both hands in both cases but it deforms the texture of the face due to the strong regularization that domains the estimation. The optical flow method only gives a



(a) Reference image (b) 6TwistFlow (c) 3DmotionALL (d) LDOF

Fig. 6.12: *Warped images for the large Z-motion experiment. The use of depth data allows a better modeled of the observed motion. Particularly, 6TwistFlow performs the most accurate estimate of the Z-motion. None of the method is able to model the large head rotation on the left side.*

roughly estimation of the observed motion. Note that none of the methods is able to estimate the head motion of the woman on the right, which presents a large rotation between both frames. Figure 6.13 shows the Z-component of the scene flow, where is evident the benefit of using the depth constraints in conjunction with a projective warp that models the image motion. The optical flow method is unable to determine the motion boundaries and suffers from the depth data noise, since changes in depth are directly estimated using the raw depth data.



(a) 6TwistFlow (b) 3DmotionALL (c) LDOF

Fig. 6.13: *Changes in depth for the large Z-motion experiment. 6TwistFlow and 3DmotionALL are able to accurately estimate the changes in depth. Conversely, the optical flow method suffers from the 2D regularization and the lack of depth data, failing to estimate the motion boundaries.*

Hand rotation

This experiment presents a hand rotating in the middle of the scene, moving quasi-rigidly, as is shown in Figure 6.14. The depth range is set to include only the surface that belongs to the arm and forearm. Figure 6.15 presents the resulting optical flow and two components of the scene flow, for each method. Regarding the optical flow, it can be noted that 3DmotionALL subtly distorts the motion field due to smoothness performed on the 3D motion field. The distortion of the motion is more evident for the optical flow method, since the regularization is done on the 2D motion field. On the other hand, thanks to the regularization done on the twist motion field, 6TwistFlow performs the most accurate estimation. The advantages of the proposed methods are more evident when observing the components of the scene flow. Unlike LDOF that estimates the X-component of the 3D motion as curve levels, 6TwistFlow and 3DmotionALL are able to accurately reflect the smooth transition brought by the 3D rotation. Similarly, the estimation of the Z-component from both methods, reveal the smooth changes in depth of the 3D surface while rotating. Contrary to the results obtained by using the optical flow estimate, which is affected by the 2D errors and the noise in depth data.

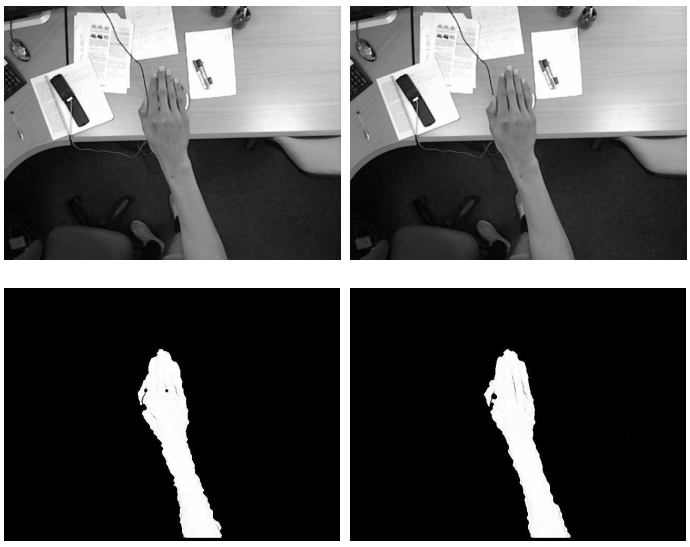
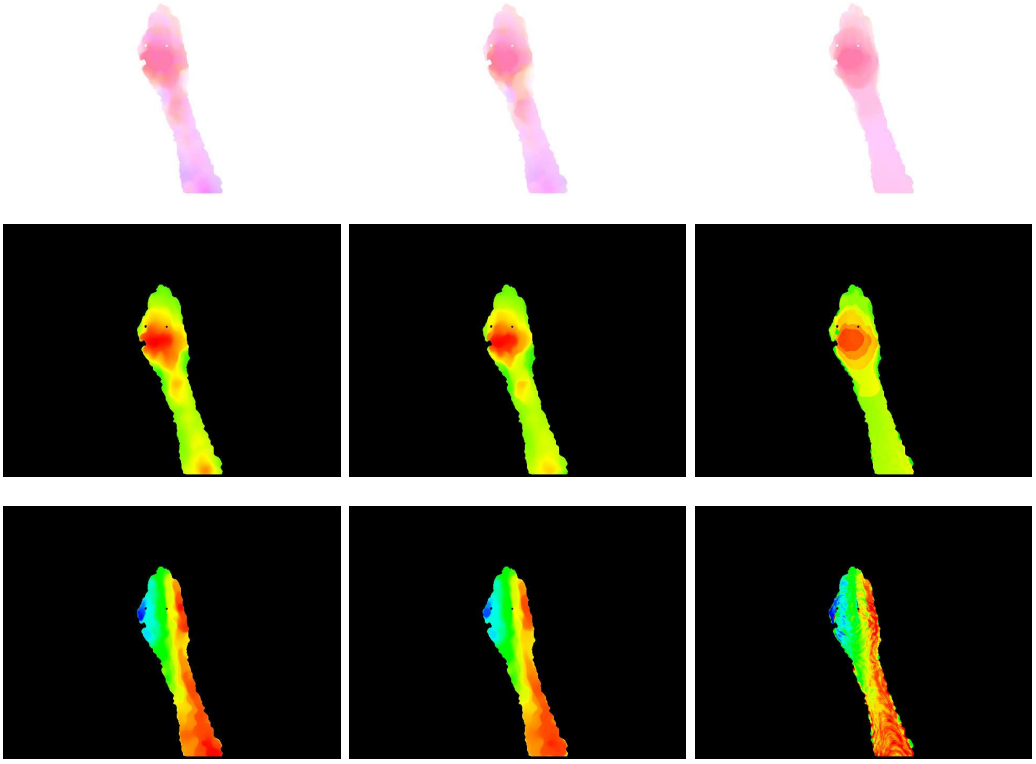


Fig. 6.14: *Inputs for the hand rotation experiment. From left to right: (a) first and (b) second pairs, of brightness and 3D structure images, respectively.*



(a) 6TwistFlow (b) 3DmotionALL (c) LDOF

Fig. 6.15: *Results of the hand rotation experiment. From top to bottom: (a) optical flow, and (b) X-component and (c) Z-component of the scene flow.*

Two arms rotation

In this experiment a fixed camera observes a motion performed with arms and hands. While hands are rotating inwards, the elbows lift, as is shown in Figure 6.16. This composite motion generates a varied optical flow, which is well estimated by our two methods, as is presented in the top images of Figure 6.17. Moreover, it can be seen that small rotations and articulated motions are well described. Observe that the optical flow method provides a smoother solution, showing be less sensitive to the specular highlights appearing on the skin. This is because its simpler 2D motion model, which is stronger regularized on the image. A higher parametrization seems to be more sensitive to this specular effect and a stronger regularization may be required. However, as is observed in the bottom images of Figure 6.17, this smoother optical flow does not imply a more accurate estimate of the scene flow. The 6TwistFlow method performs the best estimation of changes in depth, being able to model the motion of thumbs, palm hands and elbows.

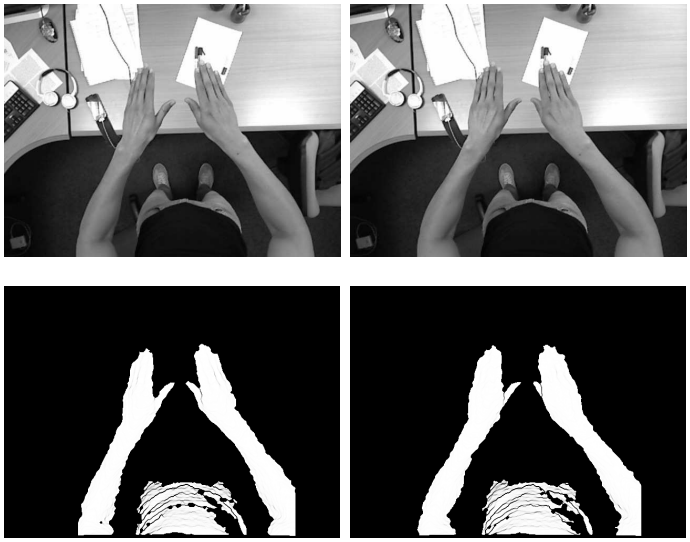
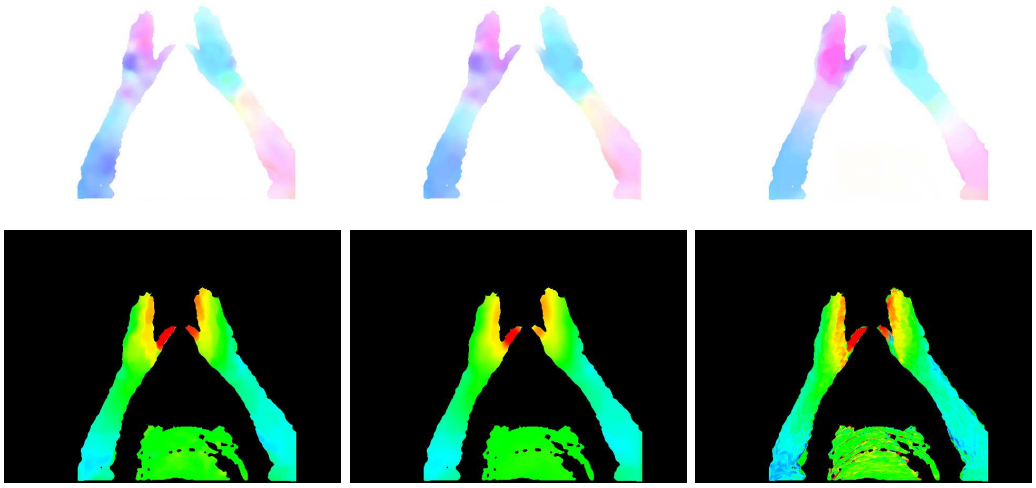


Fig. 6.16: *Inputs for the two arms rotation experiment. From left to right: (a) first and (b) second image pairs, of brightness and 3D structure.*



(a) 6TwistFlow (b) 3DmotionALL (c) LDOF

Fig. 6.17: *Results of the two arms rotation experiment. From top to bottom: optical flow and Z-component of the scene flow. LDOF provides a smoother optical flow estimation, but it does not correspond with the most accurate scene flow estimate. On the other hand, 6TwistFlow is more sensitive to the small rotation and deformation, and is able to model the motion of thumbs, palm hands and elbows. 3DmotionALL also provides an accurate estimation.*

Poster stretching

This experiment presents the stretching of a poster, as is shown in Figure 6.18. This is a challenging motion for the semi-rigid assumption due to the non-uniform deformation of the 3D surface. Figure 6.19 presents the resulting optical flow and Z-component of the scene flow. The two proposed methods are able to capture the poster deformation, thus it is possible to accurately estimate the changes in depth when the poster is folded. The gradient constancy constraint plays an important role here, since the sensor applies an automatic white balancing. Observe that the smoothness of the optical flow, inside the poster, varies according to the motion representation where the regularization is applied. Also, as can be seen on the left side of the poster, the partial occlusion brought by the stretching is estimated in a different way for each approach. Particularly, 6TwistFlow and 3DmotionALL suffer from the misalignment between the poster edges in brightness and depth data, and from the depth uncertainty near to the object boundaries. On other hand, LDOF estimates a sharp motion discontinuity since it does not have to explain the ambiguous observed data and thanks to the good contrast between the poster and the background. However, depth changes are better captured by our two methods, as is shown in the bottom images of Figure 6.19. Particularly, 3DmotionALL achieves the best performance in this experiments, accurately estimating the Z-motion and dealing in a better way with the partial occlusion, as is shown in the resulting warped images in Figure 6.20. Note that we do not use explicit occlusion awareness, but the robust norm is able to reduce its effect.

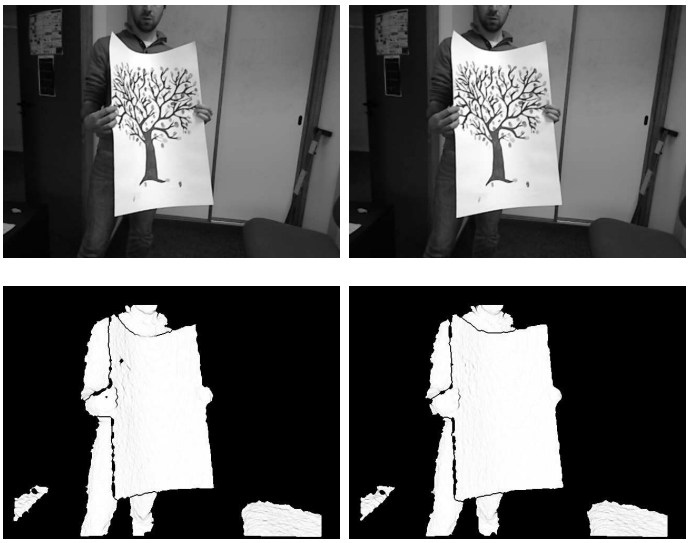
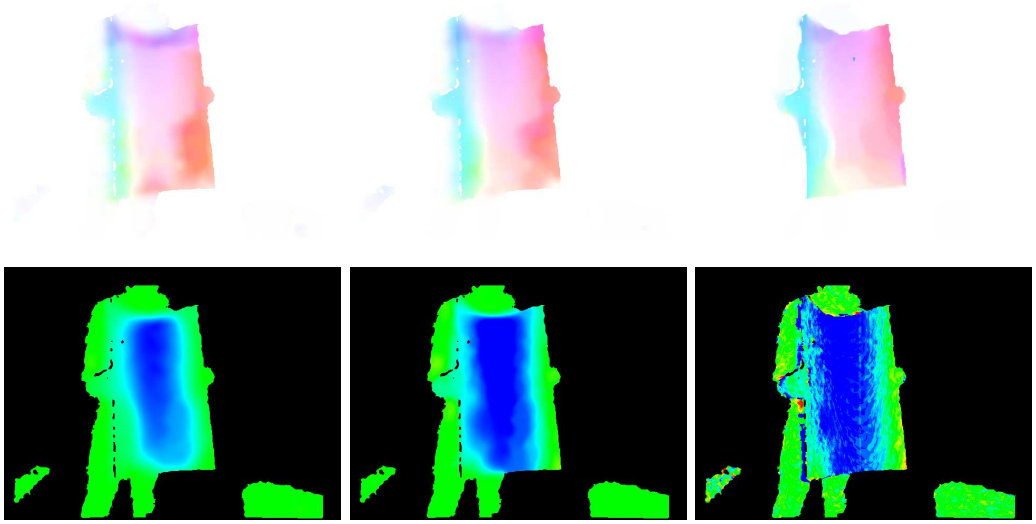
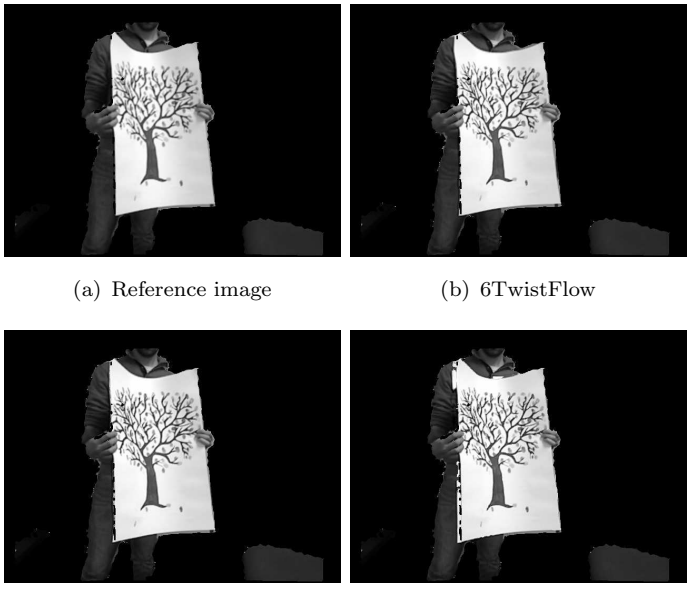


Fig. 6.18: *Inputs for the poster stretching experiment. From left to right: (a) first and (b) second image pairs, of brightness and 3D structure.*



(a) 6TwistFlow (b) 3DmotionALL (c) LDOF

Fig. 6.19: *Results of the poster stretching experiment. From top to bottom: optical flow and Z-component of the scene flow. 3DmotionALL achieves the best performance in this experiments.*



(a) Reference image (b) 6TwistFlow

(c) 3DmotionALL (d) LDOF

Fig. 6.20: *Warped images for the poster stretching experiment. Unlike LDOF, 6TwistFlow and 3DmotionALL are able to correctly warp the poster. However, observe that these two methods expand the shirt sleeve close the poster, due to the partial occlusion. In order to reduce these artifacts is required to include an explicit occlusion awareness in the estimation.*

Teddy bear

This experiment considers a pair of images of the RGBD dataset (Sturm *et al.*, 2012), where a moving camera observes a rigid scene. Figure 6.21 shows the input images. Unlike Middlebury stereo datasets, the observed motion in this experiment has a nonzero rotation component and therefore the scene flow is not constant. Optical flow results are presented in Figure 6.22, and as is expected because its motion representation, 6TwistFlow performs the most accurate estimation.

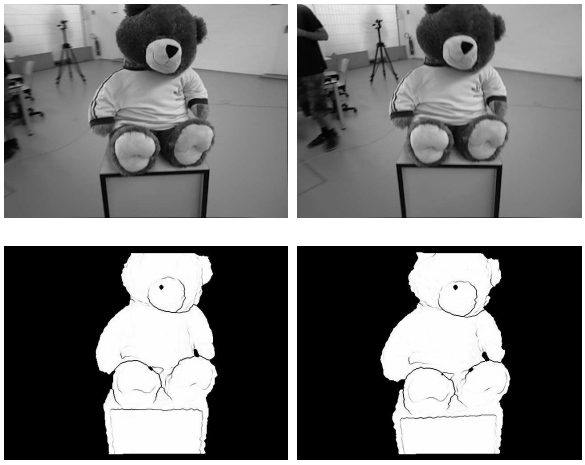


Fig. 6.21: *Inputs for the teddy bear experiment. From left to right: (a) first and (b) second image pairs, of brightness and 3D structure.*

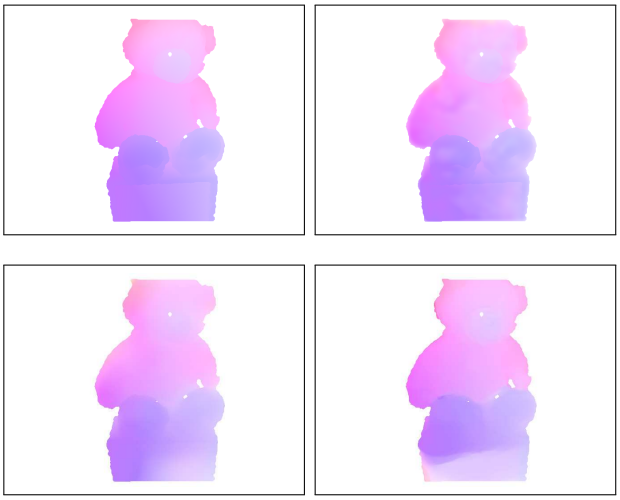


Fig. 6.22: *Optical flow results of the Teddy bear experiment. From Left to right, top to bottom: (a) rigid estimation, (b) 6TwistFlow, (c) 3DmotionALL and (d) LDOF.*

6.2.2 Rigid motion estimation

The estimation of a single rigid motion is a particular case of the proposed framework (5.2) and can be done by minimizing the rigid energy defined by 5.65. Using this formulation is possible to solve for the rigid motion that best explains both intensity and depth. For example, for the camera motion shown by the images in Figure 6.23, we set $w_I(\mathbf{x}) = 1$ and $w_Z(x) = \lambda = 0.1$, to obtain the optical flow and Z-motion presented in Figure 6.24. In this case, parameter λ allows to control the balance between brightness and depth constraints in the rigid energy. Also, it is possible to use only brightness or depth constraints, by setting $\lambda = 0$ or $\lambda = \infty$, respectively. A comparison of the final residuals of brightness and depth, for different values of λ, is presented in Figure 6.25.

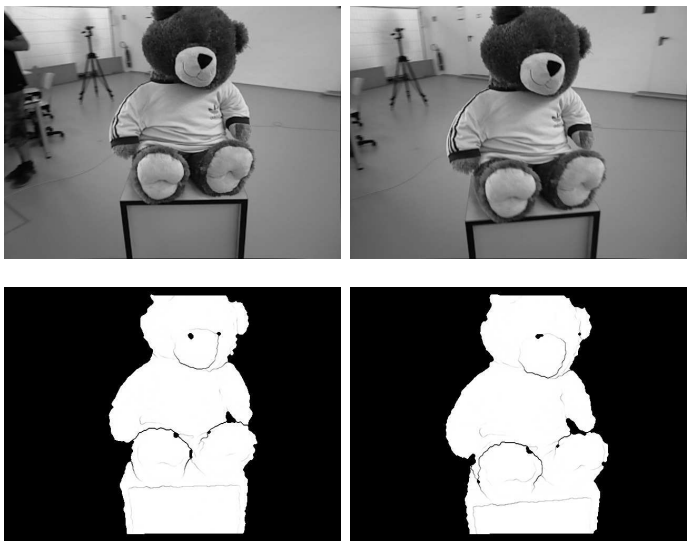
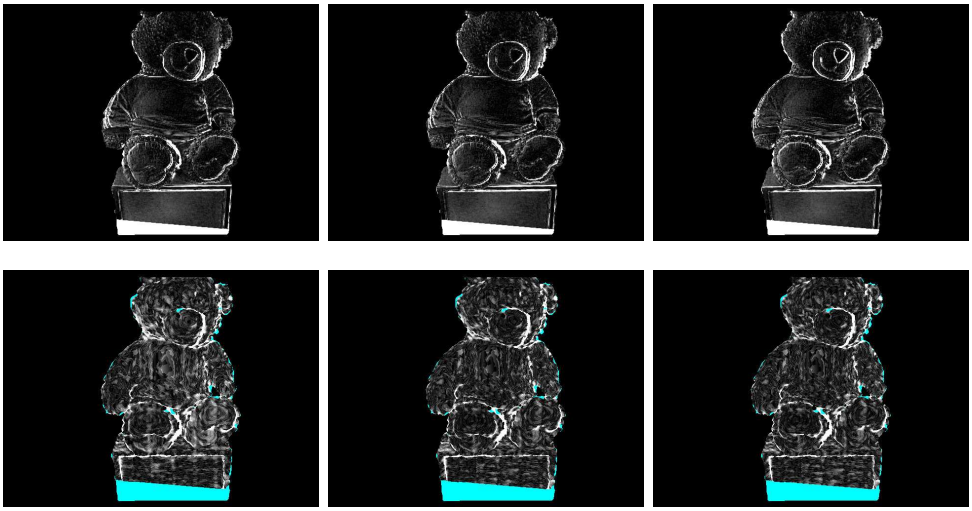


Fig. 6.23: *Inputs for the rigid motion experiment. From left to right: (a) first and (b) second image pairs, of brightness and 3D structure.*

Fig. 6.24: *Results of the rigid motion experiment. From left to right: (a) optical flow and (b) Z-component of the scene flow, respectively.*



(a) $\lambda = 0$ (b) $\lambda = 0.1$ (c) $\lambda = \infty$

Fig. 6.25: *Brightness and depth residual while varying λ. Brightness and depth residuals are presented in top and bottom images, respectively. For $\lambda = 0$, only the brightness constraints are used and the depth residual presents higher values, as is expected. On the other hand, if $\lambda = \infty$, only the depth constraints are used, and the depth residuals achieve their minimum while brightness residuals becomes large. Setting $\lambda = 0.1$ balances the contribution of brightness and depth constraints, jointly minimizing both brightness and depth residuals.*

3D reconstruction

The optimal balance between brightness and depth is an open question and should be set according to the confidence in the observed measures. Also, this balance can be defined point-wise, as we proposed in our rigid energy. Throughout experimentation we found out that setting $\lambda = 0.1$, for all pixels, allows accurate estimation using the Kinect and Xtion sensors. In order to illustrate the precision of the rigid estimation we perform a 3D reconstruction using both sensors. Getting a 3D model from a single depth image is not accurate due to noise and occlusions, but adding a couple of warped views is possible to improve the quality of the reconstruction. Accordingly, to complete the 3D model, we choose a reference frame and warp a number of close depth images using the rigid estimations. The reconstruction is done using the *Volumetric Range Image Processing Package* (VripPack) by Curless and Levoy (1996), which merges range images into a compressed volumetric grid. Experiments using images from the Xtion and Kinect sensor are presented in Figures 6.26 and 6.27, respectively. Finally, we compare the quality of the reconstruction while varying the balance between brightness and depth. In this experiment a fixed camera observes a person seated on a swivel chair, whom is requested for rotating in front of the camera, as rigid as possible. The reference frames and limit images are shown in Figure 6.28, giving an idea of the

visible surface. Observe that the considered images belong to a large rotation and therefore some parts of the reference surface result partially occluded. Moreover, the illumination is not uniform and there exist specular highlights, especially on the skin face. In Figure 6.29 the reconstructions for different values of λ are compared. In this challenging experiment, the best results are achieved using only the depth constraints.

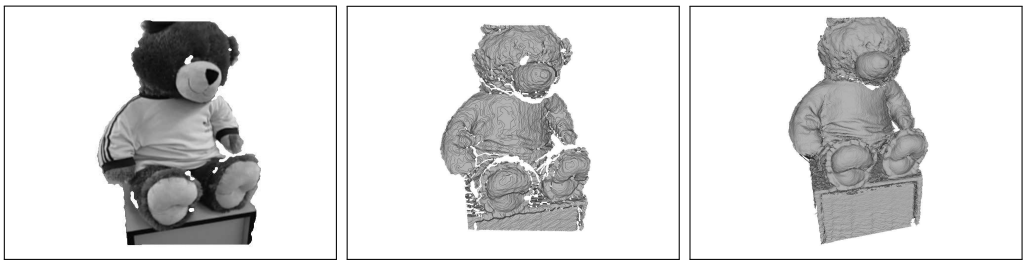


Fig. 6.26: *Teddy bear 3D reconstruction, images from the RGBD dataset (Sturm et al., 2012). From left to right: (a) reference image, and 3D reconstruction using (b) the reference depth image and (c) by warping* 12 *close views in addition to the reference depth image.*

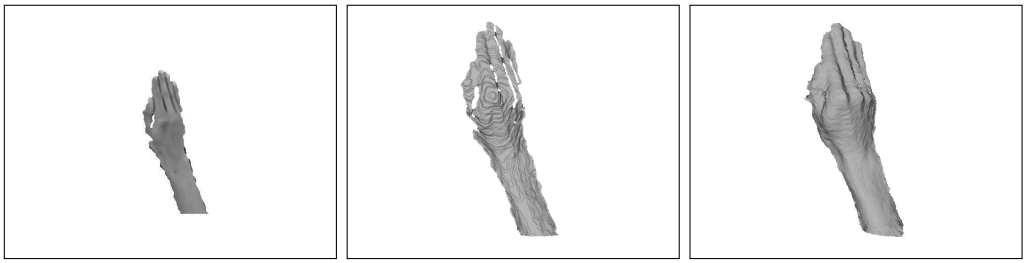
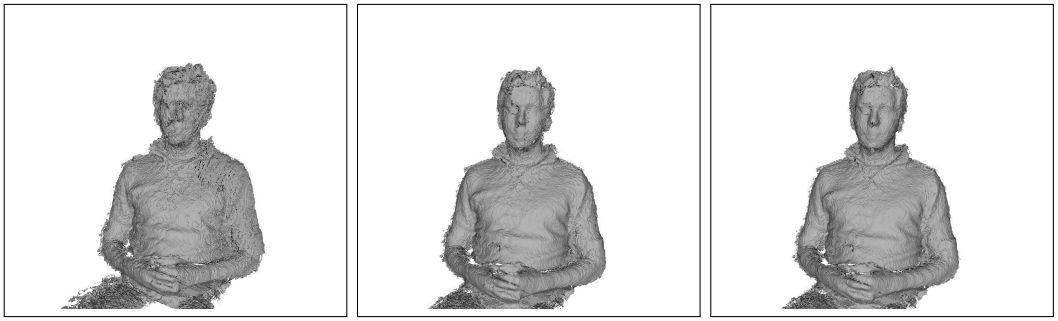


Fig. 6.27: *Rotating hand 3D reconstruction. From left to right: (a) reference image, and 3D reconstruction using (b) the reference depth image and (c) by warping* 10 *close views in addition to the reference depth image.*

(a) First frame (b) Reference frame (c) Last frame

Fig. 6.28: *Reference and limit frames for the 3D reconstruction experiment. Brightness and 3D structure images are presented in top and bottom images, respectively.*



(a) $\lambda = 0$ (b) $\lambda = 0.1$ (c) $\lambda = \infty$

Fig. 6.29: *Varying λ in the 3D reconstruction experiment. The only use of brightness constraints is highly affected by specular highlights, occlusions and the large rotation. On the other hand, the formulation of constraints only on the depth images, improves the quality of the 3D reconstruction. The jointly use of brightness and depth constraints, outperforms the only-brightness estimation but is less accurate than the results by the only-depth approach.*

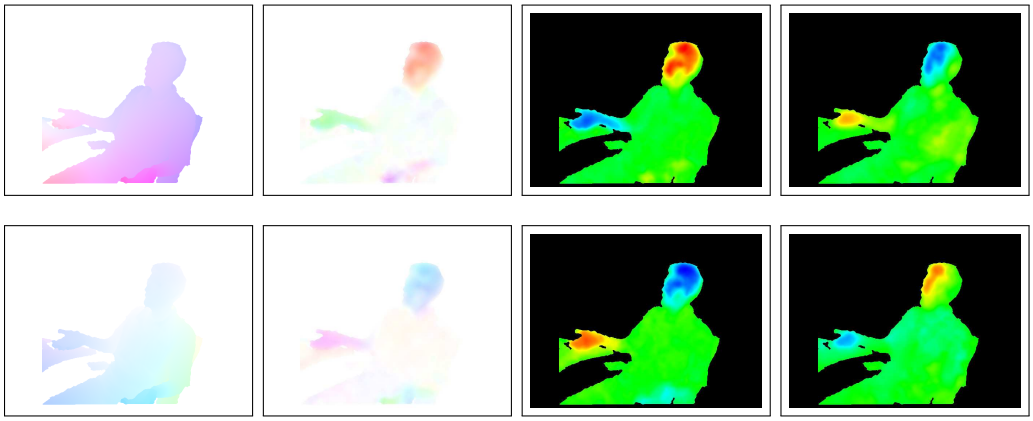
6.2.3 Nonrigid plus rigid motion estimation

In many applications, the sensor itself moves relative to the observed scene and compensating for the motion of the camera can simplify the estimation and regularization of the scene flow, as is shown in Figure 5.7 . Moreover, for 3D reconstruction of deformable objects, the camera motion is needed to register partial 3D reconstructions. Using the energy 5.64 we are able to split the motion of the scene into a globally rigid component, $\xi_{\mathrm{R}} = (\omega_{\mathrm{R}}, \tau_{\mathrm{R}}) \in \mathbb{R}^6$, capturing the camera motion relative to the dominant object/background, plus a nonrigid residual field, $\xi = (\omega, \tau)$, as we demonstrate in the two following two examples.

In the first experiment, we use the Xtion sensor to capture the motion of head and hands performed by a person, as is shown Figure 6.30. We compute the relative motion from a given reference frame, to a previous and a posterior frames. Components of the estimated motion are presented in Figure 6.31. Particularly, the motion of head and right hand are well described by the nonrigid component. On the other hand, the estimation of the left hand motion is a challenge, especially from the reference frame to the first frame, due to the sudden deformation and partial occlusion. Using the estimated scene flow, the first and last frames can be warped to the reference frame, as we done in Figure 6.32. Observe that despite the limited resolution of the RGBD images, it is possible to register the observations even under this nonrigid motion. This can be useful for constructing partial 3D models of deformable objects using a moving camera. Moreover, occlusions across

(a) First frame (b) Reference frame (c) Last frame

Fig. 6.30: *Inputs for the rigid plus nonrigid scene flow estimation using the Xtion sensor. Brightness and 3D structure images are presented in top and bottom images, respectively. While the camera is rotating to the right, the head rotates to the left and hands moves in opposite directions, one becoming partially occluded.*



(b) Rigid optical flow (c) Nonrigid optical flow (d) Nonrigid X-component (e) Nonrigid Z-component

Fig. 6.31: *Results of the rigid plus nonrigid scene flow estimation using the Xtion sensor.*

(a) First frame warped (b) Reference frame (c) Last frame warped

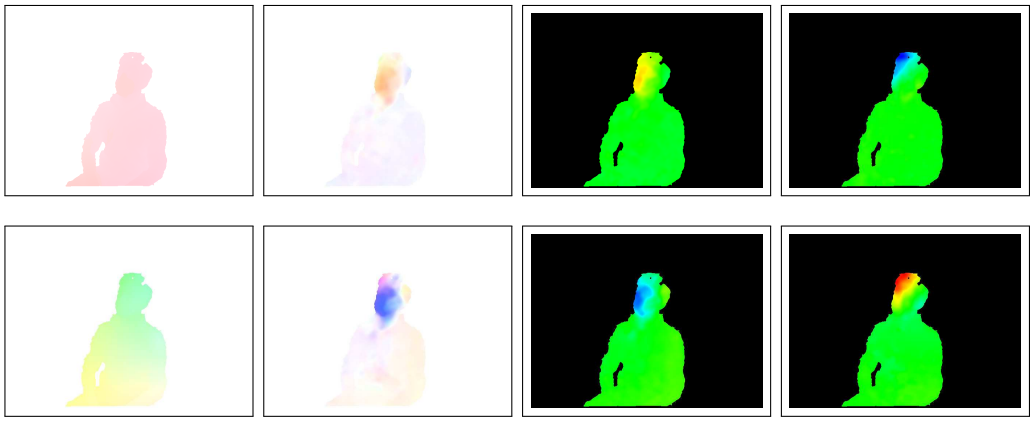
Fig. 6.32: *Warped images for the rigid plus nonrigid scene flow estimation using the Xtion sensor. Observe that the rigid motion correctly compensates for the body motion. The nonrigid estimation is able to roughly register the rotating head, as is noted by observing the right location of glasses, the beard and the ear. The motion estimation of the left hand is not well modeled.*

views can be estimated observing the depth and brightness residuals, as we illustrate in the following experiment.

In the second experiment, a Kinect sensor is subtly moved to observe the motion of the head of a person, as is shown in Figure 6.33. Components of the estimated motion are presented in Figure 6.34, where can be observed that the motion of the camera and head are well estimated. Figure 6.35 presents the resulting warped images using the estimated scene flow. Note that the head rotation is correctly compensated except for the occluded regions. By observing the brightness and depth residuals is possible to determine which regions of the reference frame are occluded in a given view, as we present in Figure 6.36. Particularly, the depth data provides confident measures to detect occlusion, unlike the brightness images, which could suffer from the specular highlights and illumination changes.

(a) First frame (b) Reference frame (c) Last frame

Fig. 6.33: *Inputs for the rigid plus nonrigid scene flow estimation using the Kinect sensor. Brightness and 3D structure images are presented in top and bottom images, respectively. While the camera is subtly moving, the head rotates and lifts.*

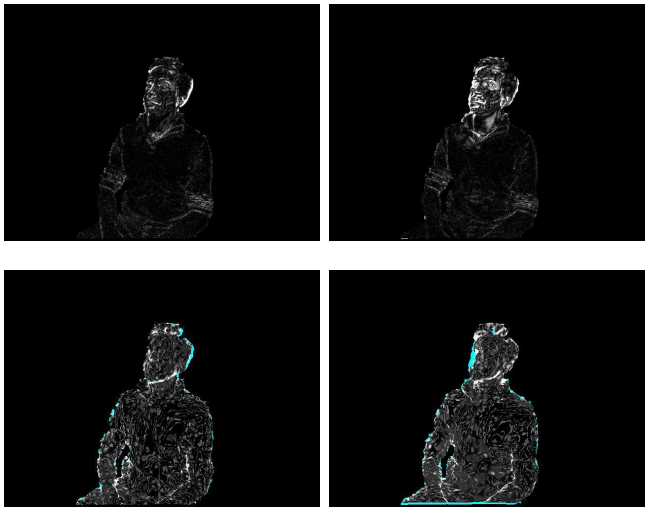


(a) Rigid optical flow (b) Nonrigid optical flow (c) Nonrigid Y-component (d) Nonrigid Z-component

Fig. 6.34: *Results of the rigid plus nonrigid scene flow estimation using the Kinect sensor.*

(a) First frame warped (b) Reference frame (c) Second frame warped

Fig. 6.35: *Warped images for the rigid plus nonrigid scene flow estimation using the Kinect sensor.*



(a) Fist frame warped (b) Second frame warped

Fig. 6.36: *Residuals of the rigid plus nonrigid scene flow estimation using the Kinect sensor. Brightness and depth residuals are presented in top and bottom images, respectively. Observe that high and undetermined residuals can be used to reliably estimate the partial occlusions. On the other hand, occlusions are also visible in the brightness residual, but can be confusable with the specular highlights which also present high residual.*

7 Conclusion

There have been two questions guiding the development of this thesis. The first question concerns the usage of both sources of data: "How to fully exploit color and depth to compute a reliably scene flow?" The second question addresses the way the motion is parametrized: "Which motion representation should be used to compute a confident scene flow?" These two questions defined the starting point of our exploration and inspired us to develop this semi-rigid framework to compute dense scene flow from RGBD images. Below we summarize the main contributions of this thesis and conclude with our vision of researching perspectives on this field.

7.1 Contributions

Color and depth exploitation

1. A warping function that allows to constrain the scene flow on RGBD images is presented. With help from the depth channel, this warp models the image motion as a fully projective function of the 3D motion. This way is possible to define consistency constraints directly on RGBD images.

2. In order to take advantage of the depth channel, we define constraints directly on the depth image, by measuring how consistent is the scene flow estimate with the observed 3D surface.

3. We simultaneously constrain the scene flow to be consistent with brightness and depth data. This allows us to exploit both sources of measurements, which are uncorrelated by nature. This way we are able to solve for the scene flow that best explains the RGBD observations.

4. The proposed framework is flexible enough to support different data terms. The set of constraints can be directly expanded, or modified, to work with

color components in any color space, and also to include constraints from invariants properties of 3D surfaces, on the depth data.

5. We have introduced different strategies to benefit from the depth channel for scene flow estimation. Depth data has been used to: *i*) model the image motion via the warping function, *ii*) directly constrain the scene flow in depth images, *iii*) guide the TV regularization, preventing smoothness across surface edges, and *iv*) estimate a prior of the local rigidity.

Motion representation

1. A semi-rigid framework for dense scene flow estimation is presented, taking advantage of the local and piecewise rigidity of most real scenes. The proposed framework supports different motion representation, data terms and regularization strategies. The evaluation demonstrates that the proposed method achieves the best results in the most commonly used scene flow benchmark.

2. We introduce an over-parametrization of the scene flow by using a field of rigid motions, with a twist motion representation. This way semi-rigid properties of the scene are directly exploited. This motion representation is provided with a warping function, allowing the formulation of RGBD constraints on the image plane.

3. Using the twist-motion representation, a semi-rigid scene flow energy is defined. Locally-rigid motions are encouraged in the data term, while piecewise rigid solutions are favor in the smoothness term. The framework provides an adjustable combination between local and piecewise rigidity.

4. We also present an alternative scene flow energy by directly using a 3D-motion representation, which is provided of a corresponding warping function. In this case, we show that the locally rigidity can be exploited in both data and smoothness terms.

5. We derive a linear version of the warp for each motion representation, enabling the solution of the scene flow by an incremental minimization. We fully describe the optimization of the two proposed energies.

6. By using the same general framework, we model the scene flow as a global rigid motion plus a nonrigid residual. This is particularly useful when estimating the motion of deformable objects in conjunction with a moving camera.

7. We have performed different experiments to evaluate some of the possibilities of the proposed framework, such as the motion representation, the brightness and depth terms balance, and the local and piecewise rigidity assumptions.

Through additional experiments we indicate the general applicability of our approach in a variety of different scenarios.

7.2 Open questions

The study done in this thesis moves forward the state of the art in scene flow by formulating a general framework for scene flow estimation. This is only a first step in a long journey of 3D motion estimation from RGBD images and our attempt opens the door to future directions for exploration.

Accuracy

Nowadays there is not an appropriate, available benchmark for scene flow. Because the 3D motion field resulting from camera translation is constant, Middlebury stereo datasets are not well suited to fully evaluate the performance of scene flow methods. Although the quality of the results on the sequences from RGBD sensors is motivating, and important remarks are done, it was not possible to measure the 3D errors for interesting nonrigid motions. Therefore, the optimal setting of parameters or the right balance between assumptions is still a query, as well as the best configuration of the proposed approach for a given specific task. For example, it is not clear the correct balance between brightness and depth terms, which could be controlling by defining pointwise confidence measures of the observed data. Moreover, further exploration is required on the appropriate amount of local and piecewise rigidity assumptions. The availability of such benchmark will allow the comparison with current methods and support the development of the new ones. It is expected that for the MPI Sintel flow dataset (Butler *et al.*, 2012), which is derived from the open source 3D animated short film Sintel, the ground truth of the depth channel be released and can be used as scene flow benchmark. Unlike Middlebury images, this dataset contains long sequences and nonrigid motions.

Depth data

Current RGBD sensors provide depth measures with a reasonable precision. For the Kinect for Xbox errors are smaller than 2mm at 1m from sensor (Khoshelham and Elberink, 2012), but increase quadratically with depth to about 4cm at the maximum distance of 5m. We have used the Charbonnier penalty to deal with the error in depth measures. However, a better modeling could be beneficial since errors in depth have a two-side influence in our approach: on the image motion modeling and on depth constancy constraints. Moreover, the quality of the registration between color and depth data is a function of errors in the depth measurement. For this reason, in some cases is evident the misalignment between color and depth,

especially close to the borders where there exists a large uncertainty on depth measures. We have used the automatic registration given by each sensor, which is based on the factory parameters, but also is possible to calibrate each sensor to improve the registration (Herrera *et al.*, 2011). Also, a preprocessing step can improve the quality of depth channel. For example, a hole-filling algorithm and a bilateral filtering jointly color images to enhace and correct the edges of the 3D surface (Richardt *et al.*, 2012).

It is important to remark that the precision, resolution and working range of depth sensors have improved in last years and is possible that some of these current issues on depth data fade away soon.

Large displacements

How to deal with large displacement is still an open question. In the rigid case, as in the Middlebury dataset, the semi-rigid assumption benefit from the constant scene flow. Nevertheless, in a nonrigid case, large displacements are harder to estimate. If the motion of smaller structures is similar to the motion of larger ones, the coarse-to-fine approach works well and improves global convergence. However, if the scale of a structure is smaller than its displacement or its motion is no coherent with larger structures, the scene flow is not well estimated. Larger structures dominate the estimation in lower scales, and then local minima in higher scales prevent the correct estimation. It is possible to include a set of sparse 2D matches in the variational formulation (Brox and Malik, 2011), e.g., using SIFT or SURF features. We follow this approach in (Quiroga *et al.*, 2013), where the scene flow is skewed to satisfy a set of 3D motion hypothesis to deal with large displacements. Nevertheless, the low quality of brightness images and the lack of texture, as presented on the skin and uniform clothes, make the matching of 2D features a challenge. Also, most of these features are located at edges of the brightness image, coinciding with edges of the 3D surface where depth measures are not reliable or available, and therefore no valid 3D motion hypothesis. For this reason, a further exploration has to be done. An interesting alternative can be to combine the proposed framework with a discrete minimization procedure to avoid get trapped in poor local optima, as is done by Lempitsky *et al.* (2008) using graph cuts. Also, it would possible to directly perturb the field of rigid motions to deal with larger motions, as is proposed by Hornacek *et al.* (2014). Finally, according to the specific application, a priori information of the scene of the deformable object can be exploited to achieve reliable estimations.

Regularization of the motion field

Total variation is of the most popular regularizers for variational motion estimation because its nice properties on discontinuities. However, its extension to functions with values in a manifold is an open problem. This is precisely the regularization

case on the field of rigid motions. We have decoupled the regularization procedure for the rotational and translational part and proposed some approximations to simplify the optimization. The decoupled regularization may not be the optimal choice from the modeling point of view since the discontinuities of both components are expected to coincide. Moreover, it is only an approximation of the real structure of the manifold. However, the decoupling allows for a simple and effective solution, and also allows the framework to support other regularization strategies. Recently, Lellmann *et al.* (2013) reformulate TV as a multi-label optimization problem with an infinite number of labels. This hard optimization problem is approximately solved using convex relaxation and is applied to regularize three-dimensional rotations. Future advances on manifold regularization may provide even more accurate and faster solvers that can be used with our parameterization. On the other hand, the regularization of the 3D motion field can be exactly formulated using TV. Nevertheless, there exist different alternatives that can be adopted and a further exploration is required. Results using channel-by-channel and vectorial TV are very similar in our experimentation. Also, it is possible to use alternative norms, as the Huber norm, or consider the total generalized variation (Bredies *et al.*, 2010). An appropriate scene flow benchmark is required to analyze in detail advantages and differences among these strategies.

7.3 Future work

Temporal consistency

Most scene flow applications use RGBD sequences as input and therefore temporal information can be exploited. Enforcing temporal consistency has proved to be useful for optical flow estimation (Volz *et al.*, 2011), and also recently, it has been used for scene flow estimation from stereo (Vogel *et al.*, 2014), achieving state-of-the-art results. Particularly, 3D motion estimation can benefit from temporal information thanks to the reluctance of objects to change their way of moving (inertia). For this reason, using a sufficiently high frame rate, real scenes present temporal consistency over more than two frames due to the inertia of moving surfaces. Accordingly, locally rigid motions can be assumed to be approximately constant on a short time interval. Also, for the 3D motion representation, smooth 3D trajectories can be encouraged in the solution. Using more than two frames yields to a more robust formulation, where the influence of noise, occlusion and missing data can be reduced.

Real-time implementation

Computational speed was not our primary concern in this work, but there are several ways to speed up a scene flow method that follows our framework. First

of all, it is highly parallelizable since computations are done independently per pixel and only considering few nearby local values. Moreover, the local rigidity is configurable and can be even reduced to one pixel to accelerate the method. Also, the image warping can be applied every n iterations instead of every iteration, an early stopping criterion can be used in the Gauss-Newton and TV solvers. Finally, depending on the application, a subsampled version of the field of rigid motions can be computed, i.e., considering only every nth pixel in the motion field, or alternatively stopping the computation in a coarser scale in the RGBD pyramid. Time benefit of some of these strategies is analyzed during our experimentation.

Scene flow descriptors

There is no work that directly computes scene flow to perform tasks such as action recognition, gesture classification or interaction. Probably, this is due to the fact that most existing methods require fully calibrated stereo or multi-view camera systems, which are not always available. Besides, most of these methods require a lot of processing time, becoming not suitable for real time applications, and their performance is limited facing nonrigid motions, which are frequent in such tasks. On the other hand, the optical flow, which is related with the scene flow projection on the image, has been successfully used in motion analysis. For example, histograms of optical flow are commonly used in state-of-the-art techniques in action recognition to construct descriptors over spatio-temporal interest points (Laptev and Lindeberg, 2003; Niebles *et al.*, 2010) and to extract 2D trajectories by tracking key-points (Messing *et al.*, 2009; Matikainen *et al.*, 2009). Furthermore, since trajectory based methods outperform other state-of-the-art approaches for action classification (Wang *et al.*, 2011), it is promising to use scene flow to capture motion information by extracting accurate 3D trajectories, which can be used to model in a better way a wider variety of motions.

Using the proposed framework we are able to solve for an accurate dense scene flow, but is also possible to solve for a set of sparse 3D trajectories, both in a more precise way than previous methods. Currently methods performing motion analysis from RBGB data, such as the pose estimation approach by Shotton *et al.* (2011), are based on a human body model, which requires a huge database for training and are constrained to the specific application. Moreover, color information is not simultaneously exploited. We differ from such approaches and aim to compute a model-less scene flow. Since we do not use any restriction on the shape of moving objects or the type of movements, our approach can be used in different tasks where 3D motion information is useful. The definition of appropriate scene flow descriptors, encoding the estimated 3D motion field, can yield better and novel applications on recognition, classification and interaction using RGBD sensors. For instance, the distribution of rigid motions, histograms of 3D trajectories, displacements or orientation, can be used as descriptors.

3D reconstruction of deformable objects

Using the proposed semi-rigid energy is possible to model the motion of the scene as a rigid component plus a non-rigid residual. This motion split is particularly useful for modeling non-rigid objects. Usually, the object of interest is captured using a moving camera, in order to observe regions that are not visible from a single view. Accordingly, the camera is moved around the 3D surface, capturing new observations to complement the desired model. However, the deformable object itself moves in a non-rigid way, and this motion has to be compensated to correctly create the 3D model. Note that this deformation of the object makes the registration task a harder problem, but at the same time, can be useful since may reveal occluded regions of the object that can be used to complete the model. By estimating the rigid and non-rigid components of the scene flow, as we propose, is possible to register close RGBD frames into a reference view to create a partial 3D model. A priori information of the object is required in order exploit the observations from other frames that are not visible for the reference view, allowing the completion of the partial 3D model. This process can be repeated for a set of reference views and partials model can be combined to have a variable 3D model of the object that varies according to the given view.

7.4 Discussion

The two questions studied in this work are inspired by a common goal, the confident estimation of scene flow for a variety of scenarios. The real-time ability of determining the 3D motion field of a scene can yield computer vision applications to take full advantage of the changing 3D world. This thesis rethinks the way 3D motion is estimated, to take the most possible advantage from current RGBD sensors. We strongly believe that the ideas developed in this work will take us a step closer to reaching this goal.

We have presented a new framework to compute dense scene flow from RGBD images by exploiting the semi-rigid properties of real world scenes. By modeling motion by a dense field of twist motions, we are able to directly encourage local and piecewise rigid solutions. We demonstrate that this motion representation allows very accurate estimations in a set of interesting experiments. Current computational tools provide alternatives to develop an implementation that will be suited for practical applications. Moreover, recent advances of optimization on manifolds will enable the exact formulation of the problem and its solution.

In order to exploit RGBD images we have developed a projective function supporting different motion representations. Using this warp is possible to directly exploit color and depth images provided by a RGBD sensor. In addition to the classical brightness-based constancy constraints, we introduce a depth constraint which allows the fully exploitation of depth data. By jointly minimizing the brightness and depth constraints we incrementally solve for the scene flow that best explains the provided images. The variational solution is directly formulated in the image domain, showing be able to accurately estimate rigid and nonrigid motions. Alternative representations of the scene can be required, particularly for 3D reconstruction applications, and discrete optimization can be used in addition to the variational formulation, to deal with larger displacement. Future advances of RGBD sensors will benefit the proposed framework, enabling access to a better quality data.

In this thesis we go beyond previous RGBD scene flow methods, by generalizing some of them and reformulating the way physical properties are used for 3D motion estimation. However, the problem scene flow estimation is not solved and there are several challenges waiting for being resolved, as some of those we have described above. Particularly, interesting challenges are posed by the requirement of measuring the confidence of components of the RGBD data and reliably detecting partial occlusions, which can take us to a closer point from a confident scene flow estimation.

Bibliography

Adiv, G. "Determining three-dimensional motion and structure from optical flow generated by several moving objects". In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:384– 401 (1985).

Anandan, P. "A computational framework and an algorithm for the measurement of visual motion". In *International Journal of Computer Vision*, 2(3):283–310 (1989). ISSN 0920-5691.

Aujol, J.F. "Some first-order algorithms for total variation based image restoration." In *Journal of Mathematical Imaging and Vision*, 34(3):307–327 (2009).

Baker, S. and Matthews, I. "Lucas-Kanade 20 years on: A unifying framework". In *International Journal of Computer Vision*, 56:221–255 (2004).

Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M. and Szeliski, R. "A database and evaluation methodology for optical flow". In "International Conference on Computer Vision", (2007).

Basha, T., Moses, Y. and Kiryati, N. "Multi-view scene flow estimation: A view centered variational approach". In "Conference on Computer Vision and Pattern Recognition", pages 1506–1513 (2010).

Bergen, J., Anandan, P., Hanna, K. and Hingorani, R. "Hierarchical model-based motion estimation". In "European Conference on Computer Vision", (1992).

Bermudez, A. and Moreno, C. "Duality methods for solving variational inequalities". In *Computers and Mathematics with Applications*, 7(1):43 – 58 (1981).

Besl, P. and McKay, N.D. "A method for registration of 3-d shapes". In *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256 (1992).

Bredies, K., Kunisch, K. and Pock, T. "Total generalized variation". In *SIAM Journal on Imaging Sciences*, 3(3):492–526 (2010).

Brox, T., Bruhn, A., Papenberg, N. and Weickert, J. "High accuracy optical flow estimation based on a theory for warping". In "European Conference on Computer Vision", (2004).

Brox, T. and Malik, J. "Large displacement optical flow: Descriptor matching in variational motion estimation." In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513 (2011).

Bruhn, A., Weickert, J. and SchnÃ¶rr, C. "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods". In *International Journal of Computer Vision*, 61(3):211–231 (2005).

Butler, D.J., Wulff, J., Stanley, G.B. and Black, M.J. "A naturalistic open source movie for optical flow evaluation". In "Proceedings of the 12th European Conference on Computer Vision - Volume Part VI", ECCV'12, pages 611–625. Springer-Verlag, Berlin, Heidelberg (2012).

Carceroni, R. and Kutulakos, K. "Multi-view scene captured by surfer sampling: From video streams to non-rigid 3D motion". In *International Journal of Computer Vision*, 110:75–90 (2008).

Chambolle, A. "An algorithm for total variation minimization and applications". In *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97 (2004).

Chambolle, A. "Total variation minimization and a class of binary mrf models". In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 136–152 (2005).

Chambolle, A. and Pock, T. "A first-order primal-dual algorithm for convex problems with applications to imaging". In *Journal of Mathematical Imaging and Vision*, 40(1):120–145 (2011).

Chan, T.F., Golub, G.H. and Mulet, P. "A nonlinear primal-dual method for total variation-based image restoration". In *SIAM J. Sci. Comput.*, 20(6):1964–1977 (1999).

Curless, B. and Levoy, M. "A volumetric method for building complex models from range images". In "Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques", SIGGRAPH '96, pages 303–312. ACM, New York, NY, USA (1996).

Devernay, F., Mateus, D. and Guilbert, M. "Multi-camera scene flow by tracking 3D points and surfels". In "Conference on Computer Vision and Pattern Recognition", (2006).

Fang, J. and Huang, T. "Solving three-dimensional small-rotation motion equations". In "Conference on Computer Vision and Pattern Recognition", (1983).

Fukurawa, Y. and Ponce, J. "Dense 3D motion capture from synchronized video streams". In "Conference on Computer Vision and Pattern Recognition", (2008).

Geiger, A., Lenz, P. and Urtasun, R. "Are we ready for autonomous driving? the kitti vision benchmark suite". In "Conference on Computer Vision and Pattern Recognition", (2012).

Goldluecke, B. and Cremers, D. "An approach to vectorial total variation based on geometric measure theory". In "Conference on Computer Vision and Pattern Recognition (CVPR)", pages 327–333 (2010).

Goldluecke, B., Strekalovskiy, E. and Cremers, D. "The natural vectorial total variation which arises from geometric measure theory." In *SIAM Journal on Imaging Sciences*, 5(2):537–563 (2012).

Green, P.J. "Iteratively Reweighted Least Squares for Maximum Likelihood Estimation, and some Robust and Resistant Alternatives". In *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(2) (1984).

Hadfield, S. and Bowden, R. "Kinecting the dots: Particle based scene flow form depth sensors". In "International Conference on Computer Vision", (2011).

Hadfield, S. and Bowden, R. "Scene particles: Unregularized particle-based scene flow estimation". In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3) (2014).

Herbst, E., Ren, X. and Fox, D. "RGB-D flow: Dense 3-D motion estimation using color and depth". In "International Conference on Robotics and Automation (ICRA)", pages 2276–2282 (2013).

Herrera, D., Kannala, J. and Heikkila, J. "Accurate and practical calibration of a depth and color camera pair". In "International Conference on Computer Analysis of Images and Patterns", (2011).

Horn, B.K. and Schunck, B.G. "Determining optical flow". In *Artificial Intelligence*, 17:185 – 203 (1981).

Horn, B. and Weldon, E. "Direct methods for recovering motion". In *International Journal of Computer Vision*, 2:51–76 (1988).

Hornacek, M., Fitzgibbon, A. and Rother, C. "Sphereflow: 6 dof scene flow from RGB-D pairs". In "Conference on Computer Vision and Pattern Recognition", (2014).

Hornacek, M., Rhemann, C., Gelautz, M. and Rother, C. "Depth super resolution by rigid body self-similarity in 3d". In "Conference on Computer Vision and Pattern Recognition (CVPR)", pages 1123–1130 (2013).

Huguet, F. and Devernay, F. "A variational method for scene flow estimation from stereo sequences". In "International Conference on Computer Vision", pages 1–7 (2007).

Kerl, C., Sturm, J. and Cremers, D. "Robust odometry estimation for RGB-D cameras". In "International Conference on Robotics and Automation (ICRA)", pages 3748–3754 (2013).

Khoshelham, K. and Elberink, S.O. "Accuracy and resolution of kinect depth data for indoor mapping applications". In *Sensors*, 12(2):1437–1454 (2012).

Laptev, I. and Lindeberg, T. "Space-time interest points". In "European Conference on Computer Vision", (2003).

Lellmann, J., Strekalovskiy, E., Koetter, S. and Cremers, D. "Total variation regularization for functions with values in a manifold". In "International Conference on Computer Vision", (2013).

Lempitsky, V., Roth, S. and Rother, C. "Fusionflow: Discrete-continuous optimization for optical flow estimation". In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 0:1–8 (2008).

Letouzey, A., Petit, B. and Boyer, E. "Scene flow from depth and color images". In "British Machine Vision Conference (BMVC), 2011", (2011).

Li, H., Adams, B., Guibas, L.J. and Pauly, M. "Robust single-view geometry and motion reconstruction". In "ACM SIGGRAPH Asia 2009 papers", pages 175:1–175:10 (2009).

Longuet-Higgins, H. and Prazdny, K. "The interpretation of a moving retinal image". In *Royal Society London*, 208:385–397 (1980).

Lucas, B.D. and Kanade, T. "An iterative image registration technique with an application to stereo vision". In "Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2", IJCAI'81, pages 674–679 (1981).

Lukins, T. and Fisher, R. "Colour constrained 4d flow". In "British Machine Vision Conference", (2005).

Matikainen, P., Hebert, M. and Sukthankar, R. "Trajectons: Action recognition through the motion analysis of tracked features". In "International Conference on Computer Vision Workshops", (2009).

Messing, R., Pal, C. and Kautz, H. "Activity recognition using the velocity histories of tracked keypoints". In "International Conference on Computer Vision, 2009", (2009).

Neumann, J. and Aloimonos, Y. "Spatio-temporal stereo using multi-resolution division surfaces". In *International Journal of Computer Vision*, pages 181–193 (2002).

Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S. and Fitzgibbon, A. "Kinectfusion: Real-time dense surface mapping and tracking". In "IEEE ISMAR", IEEE (2011).

Niebles, J., Chen, C. and Fei-Fei, L. "Modelling temporal structure of decomposable motion segments for activity classification". In "European Conference on Computer Vision", (2010).

Nir, T., Bruckstein, A.M. and Kimmel, R. "Over-parameterized variational optical flow". In *International Journal on Computer Vision*, 76(2):205–216 (2008).

Quiroga, J., Brox, T., Devernay, F. and Crowley, J. "Dense semi-rigid scene flow estimation from rgbd images". In "European Conference on Computer Vision", (2014a).

Quiroga, J., Devernay, F. and Crowley, J. "Scene flow by tracking in intensity and depth data". In "Computer Vision and Pattern Recognition Workshops (CVPRW)", pages 50–57 (2012).

Quiroga, J., Devernay, F. and Crowley, J. "Local/global scene flow estimation". In "International Conference on Image Processing (ICIP)", pages 3850–3854 (2013).

Quiroga, J., Devernay, F. and Crowley, J. "Local scene flow by tracking in intensity and depth". In *Journal of Visual Communication and Image Representation*, 25(1):98 – 107 (2014b).

Richardt, C., Stoll, C., Dodgson, N.A., Seidel, H.P. and Theobalt, C. "Coherent spatiotemporal filtering, upsampling and rendering of rgbz videos." In *Comput. Graph. Forum*, 31(2):247–256 (2012).

Rosman, G., Bronstein, A., Bronstein, M., Tai, X.C. and Kimmel, R. "Group-valued regularization for analysis of articulated motion". In "European Conference on Computer Vision Workshops", pages 52–62 (2012).

Rudin, L.I., Osher, S. and Fatemi, E. "Nonlinear total variation based noise removal algorithms". In "Proceedings of the Eleventh Annual International Conference of the Center for Nonlinear Studies on Experimental Mathematics", pages 259–268. Elsevier North-Holland, Inc., Amsterdam, The Netherlands, The Netherlands (1992).

Scharstein, D. and Szeliski, R. "High-accuracy stereo depth maps using structured light". In "Conference on Computer Vision and Pattern Recognition", pages 195–202 vol.1 (2003).

Shi, J. and Tomasi, C. "Good features to track". In "Conference on Computer Vision and Pattern Recognition, 1994", (1994).

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A. and Blake, A. "Real-time human pose recognition in parts from a single depth image". In "Conference on Computer Vision and Pattern Recongintion", IEEE (2011).

Spies, H., Jahne, B. and Barron, J. "Dense range flow from depth and intensity data". In "International Conference on Pattern Recognition", pages 131–134 vol.1 (2000).

Sturm, J., Engelhard, N., Endres, F., Burgard, W. and Cremers, D. "A benchmark for the evaluation of RGB-D slam systems". In "International Conference on Intelligent Robot Systems (IROS)", pages 573–580 (2012).

Sun, D., Roth, S. and Black, M. "Secrets of optical flow estimation and their principles". In "Conference on Computer Vision and Pattern Recognition (CVPR)", pages 2432–2439 (2010).

Valgaerts, L., Bruhn, A., Zimmer, H., Weickert, J., Stoll, C. and Theobalt, S. "Joint estimation of motion, structure and geometry from stereo sequences". In "European Conference on Computer Vision", (2010).

Vedula, S., Baker, S., Rander, P. and Collins, R. "Three-dimensional scene flow". In "International Conference on Computer Vision", pages 722–729 vol.2 (1999).

Vedula, S., Baker, S., Render, P., Collins, R. and Kanade, T. "Three-dimensional scene flow". In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:275–280 (2005).

Vogel, C., Schindler, K. and Roth, S. "3D scene flow estimation with a rigid motion prior". In "International Conference on Computer Vision", pages 1291–1298 (2011).

Vogel, C., Roth, S. and Schindler, K. "View-consistent 3d scene flow estimation over multiple frames". In D. Fleet, T. Pajdla, B. Schiele and T. Tuytelaars, editors, "European Conference on Computer Vision", volume 8692 of *Lecture Notes in Computer Science*, pages 263–278. Springer International Publishing (2014).

Vogel, C., Schindler, K. and Roth, S. "Piecewise rigid scene flow". In "International Conference on Computer Vision", pages 1377–1384 (2013).

Volz, S., Bruhn, A., Valgaerts, L. and Zimmer, H. "Modeling temporal coherence for optical flow". In "Computer Vision (ICCV), 2011 IEEE International Conference on", pages 1116–1123 (2011).

Wang, H., Kläser, A., Schmid, C. and Liu, C.L. "Action recognition by dense trajectories". In "Conference on Computer Vision & Pattern Recognition, 2011", (2011).

Wang, Y., Yang, J., Yin, W. and Zhang, Y. "A new alternating minimization algorithm for total variation image reconstruction". In *SIAM J. Img. Sci.*, 1(3):248–272 (2008).

Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U. and Cremers, D. "Efficient dense scene flow from sparse of dense stereo data". In "European Conference on Computer Vision", pages 739–751 (2008).

Wedel, A., Pock, T., Zach, C., Bischof, H. and Cremers, D. "An improved algorithm for TV-L1 optical flow". In "Statistical and Geometrical Approaches to Visual Motion Analysis", volume 5604 of *Lecture Notes in Computer Science*, pages 23–45. Springer Berlin Heidelberg (2009).

Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D. and Bischof, H. "Anisotropic Huber-L1 Optical Flow". In "Proceedings of the British Machine Vision Conference (BMVC)", (2009).

Xu, L., Jia, J. and Matsushita, Y. "Motion detail preserving optical flow estimation". In *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(9):1744–1757 (2012).

Zhang, Y. and Kambhamettu, C. "On 3D scene flow and structure estimation". In "Conference on Computer Vision and Pattern Recognition", (2001).

Zhu, M. and Chan, T. "An efficient primal-dual hybrid gradient algorithm for total variation image restoration". In *UCLA CAM Report*, pages 08–34 (2008).

Zhu, M., Wright, S.J. and Chan, T.F. "Duality-based algorithms for total-variation-regularized image restoration". In *Computational Optimization and Applications*, 47(3):377–400 (2010).

List of Figures

List of Tables

List of Algorithms